

```

0001      IDT      'MT358F10'
0002      OPTION XREF
0003      *      Jef Winsor
0004      *      3.58MHz, Full Expansion, IBC at P110
0005      *      MICROTAPe PERIPHERAL CONTROL SOFTWARE
0006      *
0007      *      SYSTEM HARDWARE IMPLEMENTATION , ----,
0008      *      , -----|SW| ADDRESS read 0 - 7
0009      *, -----, |-----|SW| SELECT add 1 for
0010      *| 70C20| | | | |JMP| device #
0011      *| | | | | | |
0012      *|04 6 A0|<--ADDR 0-----| | | | | |
0013      *| 7 A1|<--ADDR 1-----| | | | |VOLTAGE|
0014      *| 8 A2|<--ADDR 2-----| | | | |CONTROL|
0015      *| 9 A3|<--*--GND | | | | | |
0016      *| 10 A4|<--' | | | | | |
0017      *| 16 A5|<--NC | | | | | LED & |
0018      *| 15 A6|<--BATTERY LEVEL-----| | | | |WP AND |
0019      *| 11 A7|<--INPUT RAW DATA-----| | | | |EOT/BOT|
0020      *| | | | | |----->|SENSORS|
0021      *|06 3 B0|-->TEST POINT | | | | | |
0022      *| 4 B1|-->WRITE ENABLE-----+---+---, | | |
0023      *| 5 B2|-->SENSORS & LED-----| | | | | |
0024      *| 37 B3|-->MOTOR-----*---+---, | | |
0025      *| 38 B4|-->A-LATCH-----, | | | | | |
0026      *| 1 B5|-->R/W-----, | | \ | | | |
0027      *| 39 B6|-->ENABLE-----, | | | | | |
0028      *| 2 B7|-->NC | | V V V | | | | |
0029      *| | | | | |-----, | | | | | |
0030      *|08 28 C0|<->DATA 0/RS---<->| | | | | |
0031      *| 29 C1|<->DATA 1-----<->|I/O BUS| | | | | |
0032      *| 30 C2|<->DATA 2-----<->| | | | | |
0033      *| 31 C3|<->DATA 3-----<->|SUPPORT| | | | | |
0034      *| 32 C4|---CS----->| | | | | |
0035      *| 33 C5|---NC | | | | | |
0036      *| 34 C6|---NC |-----+---+---+---' | | |
0037      *| 35 C7|---NC | | +---+---+---+---' | | |
0038      *| | | | | |-----, | | | | | |
0039      *|0A 27 D0|-->WRITE DATA-----+---+---, | | | |
0040      *| 26 D1|<--EOT/BOT-----| | | | | |
0041      *| 24 D2|<--WP-----| | V | V V | | |
0042      *| 23 D3|<-- ---*--- +V ROM | |-----, | | |
0043      *| 22 D4|<-- ---*--- OPTIONS | | | | |TAPE DRIVE|
0044      *| 21 D5|<-- ---* | | | | | |
0045      *| 20 D6|<-- ---' | | | | | |
0046      *| 19 D7|<--NC | | | | | |
0047      *| | | | | |-----, | | | | | |
0048      *| 13 INT1|<--IRQ-----' | | | | | |
0049      *| 12 INT3|<--TAPE TRANSITION PULSE-----' | | | | |
0050      *| | | | | |-----, | | | | | |

```

| | | | |
|------|-------------------------|--------|-------------------|
| 0052 | ----- | | |
| 0053 | * register file equates | | |
| 0054 | 0000 | AREG | EQU R0 |
| 0055 | 0001 | BREG | EQU R1 |
| 0056 | 0002 | BITCON | EQU R2 |
| 0057 | 0003 | STACK | EQU R3 |
| 0058 | 0013 | STACKL | EQU >13 |
| 0059 | 0016 | REGS | EQU STACK+STACKL |
| 0060 | 000C | NMLEN | EQU >0C |
| 0061 | 002D | SAB | EQU REGS+11+NMLEN |
| 0062 | 0016 | FNAMEL | EQU SAB-11-NMLEN |
| 0063 | 0021 | FNAME1 | EQU SAB-12 |
| 0064 | 0022 | ATTRIB | EQU SAB-11 |
| 0065 | 0024 | BOPEN | EQU SAB-9 |
| 0066 | 0026 | DLEN | EQU SAB-7 |
| 0067 | 0028 | BLEN | EQU SAB-5 |
| 0068 | 002A | RNUM | EQU SAB-3 |
| 0069 | 002B | LUNO | EQU SAB-2 |
| 0070 | 002C | CCODE | EQU SAB-1 |
| 0071 | 002D | DCODE | EQU SAB |
| 0072 | 002D | STATUS | EQU DCODE |
| 0073 | 0029 | TEMP4 | EQU RNUM-1 |
| 0074 | 002B | TEMP2 | EQU LUNO |
| 0075 | 002C | TEMP1 | EQU CCODE |
| 0076 | 002F | NIBCON | EQU SAB+2 |
| 0077 | 002F | COUNT | EQU NIBCON |
| 0078 | 0031 | CHKSUM | EQU NIBCON+2 |
| 0079 | 0032 | INT2 | EQU CHKSUM+1 |
| 0080 | 0034 | INT2V | EQU CHKSUM+3 |
| 0081 | 0036 | DATAP | EQU INT2V+2 |
| 0082 | 0037 | FFLAG | EQU DATAP+1 |
| 0083 | 0039 | RECFIL | EQU FFLAG+2 |
| 0084 | 003B | MAXLEN | EQU RECFIL+2 |
| 0085 | 003D | NREC | EQU MAXLEN+2 |
| 0086 | 003E | NFILE | EQU NREC+1 |
| 0087 | 003F | DIRECT | EQU R63 |
| 0088 | 0040 | FILEO | EQU R64 |
| 0089 | 0044 | FILE1 | EQU R68 |
| 0090 | 0048 | FILE2 | EQU R72 |
| 0091 | 004C | FILE3 | EQU R76 |
| 0092 | 0050 | FILE4 | EQU R80 |
| 0093 | 0054 | FILE5 | EQU R84 |
| 0094 | 0058 | FILE6 | EQU R88 |
| 0095 | 005C | FILE7 | EQU R92 |
| 0096 | 0060 | FILE8 | EQU R96 |
| 0097 | 0064 | FILE9 | EQU R100 |
| 0098 | 0068 | FILEA | EQU R104 |
| 0099 | 006C | FILEB | EQU R108 |
| 0100 | 0070 | FILEC | EQU R112 |
| 0101 | 0074 | FILED | EQU R116 |
| 0102 | 0078 | FILEE | EQU R120 |
| 0103 | 007C | FILEF | EQU R124 |
| 0104 | 007F | ENDRCT | EQU R127 |

a register
 b register
 nibble bit-counter
 soft stack location
 reserved stack length
 start of registers
 file name length
 ms address byte of SAB
 last character of FILENA
 first character of FILEN
 attributes byte
 BL data from OPEN
 data length
 buufer length
 record number
 logical unit number
 command code
 device code
 status to be returned
 temporary use
 temporary use
 temporary use
 DL nibble counter
 16 bit counter
 checksum value (16 bits)
 INT2 entry point (m code)
 INT2 ram-vector
 data pointer
 drive status flags
 number of records
 maximum record length
 current record number
 current file number
 start of ram directory
 start of FILEO status
 start of FILE1 status
 start of FILE2 status
 start of FILE3 status
 start of FILE4 status
 start of FILE5 status
 start of FILE6 status
 start of FILE7 status
 start of FILE8 status
 start of FILE9 status
 start of FILEA status
 start of FILEB status
 start of FILEC status
 start of FILED status
 start of FILEE status
 start of FILEF status
 end of directory

0106 *-----
0107 COPY ALC. STRINGY. SRC. P110
A0001 * peripheral file equates
A0002 0000 IOCNTL EQU P0 i/o control
A0003 0002 TIME EQU P2 timer start value
A0004 0003 CAPTUR EQU P3 timer value @ INT34
A0005 0003 PSSCALE EQU P3 prescale control
A0006 0003 TIMER EQU P3 timer control
A0007 0004 TEST EQU P4 data & system test port
A0008 0006 DRIVE EQU P6 drive & system control
A0009 000A WAFER EQU P10 wafer i/o
A0010 000B DDRD EQU P11 ddr for PORT D
A0011 0110 BDATA EQU >110 bus data
A0012 0111 BCNTL EQU >111 bus control
A0013 0111 BSTAT EQU >111 bus status
0108 * COPY ALC. STRINGY. SRC. P180

```
0110      *-----  
0111      * flag use descriptions  
0112      *-----  
0113      * FFLAG      file management flags  
0114      *-----  
0115      * Value!! 1 ! 0  
0116      *-----  
0117      * Bit 7!!at EOF !not at EOF  
0118      *-----  
0119      * Bit 6!!error !no error  
0120      *-----  
0121      * Bit 5!!protect!no protect  
0122      *-----  
0123      * Bit 4!!open !no open  
0124      *-----  
0125      * Bit 3!!found !not found  
0126      *-----  
0127      * Bit 2!!at EOT !not at EOT  
0128      *-----  
0129      * Bit 1!!filename!number  
0130      *-----  
0131      * Bit 0!! --- ! ---  
0132      *-----  
0133      *  
0134      *  
0135      *  
0136      *-----  
0137      * RECFIL-1      file parameter flags  
0138      *-----  
0139      * Value!! 1 ! 0  
0140      *-----  
0141      * Bit 7!!active !inactiv  
0142      *-----  
0143      * Bit 6!!last !not 1st  
0144      *-----  
0145      * Bit 5!! --- ! ---  
0146      *-----  
0147      * Bit 4!!intrnal!display  
0148      *-----  
0149      * Bit 3!! --- ! ---  
0150      *-----  
0151      * Bit 2!! --- ! ---  
0152      *-----  
0153      * Bit 1!! --- ! ---  
0154      *-----  
0155      * Bit 0!! --- ! ---  
0156      *-----
```

```
0158      *-----  
0159      * ATTRIB      file attribute flags  
0160      *-----  
0161      * Value!! 1  !  0  
0162      *-----  
0163      * Bit 7!!out/upd!app/inp  
0164      *-----  
0165      * Bit 6!!inp/upd!app/out  
0166      *-----  
0167      * Bit 5!!relative/sequent  
0168      *-----  
0169      * Bit 4!!fixed !variable  
0170      *-----  
0171      * Bit 3!!internal!display  
0172      *-----  
0173      * Bit 2!!  res  !  res  
0174      *-----  
0175      * Bit 1!!  res  !  res  
0176      *-----  
0177      * Bit 0!!  res  !  res  
0178      *-----  
0179      *  
0180      *  
0181      *  
0182      *-----  
0183      * DIRECT      directory data  
0184      *-----  
0185      * Value!! 1  !  0  
0186      *-----  
0187      * Bit 7!!revision # bit 3  
0188      *-----  
0189      * Bit 6!!revision # bit 2  
0190      *-----  
0191      * Bit 5!!revision # bit 1  
0192      *-----  
0193      * Bit 4!!revision # bit 0  
0194      *-----  
0195      * Bit 3!!  ---  !  ---  
0196      *-----  
0197      * Bit 2!!  ---  !  ---  
0198      *-----  
0199      * Bit 1!!  ---  !  ---  
0200      *-----  
0201      * Bit 0!!  ---  !  ---  
0202      *-----
```

```

0204      *-----*
0205      * wafer drive control constant equates
0206      00FF  INITDR EQU  >FF           initialize drive
0207      0001  INITWF EQU  >01           initialize wafer dör
0208      00F3  MT     EQU  >F3           turn on motor & sensor
0209      0008  MTBAR EQU  >08           turn off motor
0210      00F1  MTWE  EQU  >F1           turn on motor, WE & sens
0211      0000  RENITW EQU  >00           tri-state DO
0212      00FB  SN    EQU  >FB           turn on EOT/BOT sensor
0213      00FB  STOP  EQU  >FB           turn off drive except se
0214      00F9  WE    EQU  >F9           turn on WE & sensor
0215      * wafer test constant equates
0216      0040  BATTRY EQU  >40           bit-test for battery-lev
0217      0002  EOTTST EQU  >02           bit-test for EOT
0218      0080  INPUT  EQU  >80           bit-test for input data
0219      0004  WP    EQU  >04           bit-test for WP
0220      * wafer data constant equates
0221      0001  INVBIT EQU  >01           wafer-write invert
0222      0008  SETBIT EQU  >08           store a "1" bit
0223      *-----*
0224      * bus control constant equates
0225      0001  DROP   EQU  >01           drop HSK bit
0226      0000  HSKSET EQU  >00           let HSK float
0227      0004  INHIB  EQU  >04           inhibit IBC
0228      0001  RELEAS EQU  >01           release-HSK bit
0229      * bus test constant equates
0230      0001  HSK   EQU  >01           bus ready bit-test
0231      0008  IRQ   EQU  >08           bus data ready bit-test
0232      0002  BAV   EQU  >02           BAV active test
0233      *-----*
0234      COPY   ALC. STRINGY. SRC. MHZ358
B0001      * timer constant equates
B0002      000D  BITIME EQU  >0D   3.58MHz, 8KBaud data half-bit time
B0003      0020  HALT   EQU  >20           stop timer
B0004      00C0  HRANGE EQU  >C0   3.58MHz
B0005      0038  LRANGE EQU  >38   3.58MHz
B0006      00BF  MAX    EQU  >BF           max timer, no sleep
B0007      0006  ORANGE EQU  >06   bit to bit compare
B0008      0000  SLEEP   EQU  >00           sleep mode
B0009      00A0  START   EQU  >A0           timer-start bit
B0010      * software loop constants
B0011      0067  BOSTIM EQU  >67   3.58MHz
B0012      00B6  BOTIME EQU  >B6   3.58MHz
0235      *-----*
0236      *-----*
0237      COPY   ALC. STRINGY. SRC. FE
C0001      * interrupt select & clear constant equates
C0002      0083  I1CS   EQU  >83           clear and select INT1
C0003      00AA  I123C EQU  >AA           clear INT1,2&3 flags
C0004      0088  I2C    EQU  >88           clear INT2 flag
C0005      008C  I2CS   EQU  >8C           clear and select INT2
C0006      0004  I2S    EQU  >04           test INT2 select bit
C0007      00A8  I23C   EQU  >A8           clear INT2&3 flags
C0008      00BC  I23CS  EQU  >BC           clear and select INT2&3
C0009      00A0  I3C    EQU  >A0           clear INT3 flag
C0010      00B0  I3CS   EQU  >B0           clear and select INT3
C0011      00B4  I3C23S EQU  >B4           clear INT3 and select IN

```

| | | | | |
|-------|------|--|-----|-------------------------------|
| 00012 | 0020 | I3FLAG EQU | >20 | test INT3 flag-bit |
| 00013 | 00A4 | RDBIT1 EQU | >A4 | read bit opcode |
| 00014 | 0020 | RDBIT2 EQU | >20 | read bit parameter |
| 00015 | 0000 | RDBIT3 EQU | >00 | read bit parameter |
| 00016 | 00D6 | WRBIT1 EQU | >D6 | write bit opcode |
| 00017 | 0002 | WRBIT2 EQU | >02 | write bit parameter |
| 0238 | * | COPY ALC. STRINGY. SRC. PE | | |
| 0239 | * | | | |
| 0240 | | * flag set & test bit constant equates | | |
| 0241 | 0080 | ACTIVE EQU | >80 | active file flag |
| 0242 | 0040 | DFORMA EQU | >40 | initialize directory |
| 0243 | 0010 | DISPLAY EQU | >10 | display flag in director |
| 0244 | 0080 | EOFFLG EQU | >80 | file is at EOF flag |
| 0245 | 0004 | EOTFLG EQU | >04 | EOT found flag |
| 0246 | 0040 | FERROR EQU | >40 | error flag |
| 0247 | 0008 | FFOUND EQU | >08 | file/record found flag |
| 0248 | 0002 | FNAME EQU | >02 | file name/number flag |
| 0249 | 0010 | FOPEN EQU | >10 | file open flag |
| 0250 | 0020 | FWP EQU | >20 | write protect flag |
| 0251 | 0000 | INIFLG EQU | >00 | initialize flags |
| 0252 | 0008 | INTDIS EQU | >08 | internal/display in attr |
| 0253 | 0010 | INTERN EQU | >10 | internal flag in directo |
| 0254 | 0040 | LAST EQU | >40 | last file flag |
| 0255 | 0000 | NEWFIL EQU | >00 | create new file |
| 0256 | 0014 | NEWFLG EQU | >14 | set up flag bits |
| 0257 | 0000 | REVO EQU | >00 | format version 0 |
| 0258 | 0020 | SEQUEN EQU | >20 | sequential file |
| 0259 | | * flag reset bit constant equates | | |
| 0260 | 00F7 | FMSSNG EQU | >F7 | file/record not found |
| 0261 | 00FD | FNUMBR EQU | >FD | file number |
| 0262 | 007F | INACTV EQU | >7F | reset active flag |
| 0263 | 00BF | LASTO EQU | >BF | reset last flag |
| 0264 | 00EF | OPNRST EQU | >EF | reset open flag |
| 0265 | 007F | RSTEOF EQU | >7F | reset EOF flag |
| 0266 | 00FB | RSTEOT EQU | >FB | reset EOT flag |
| 0267 | 00BF | RSTERR EQU | >BF | reset error flag |
| 0268 | | *RSTWRIT EQU | >FD | reset write flag |
| 0269 | 009B | WPERET EQU | >9B | reset WP, error&EOT flags |
| 0270 | * | | | |
| 0271 | | * access mode test constant equates | | |
| 0272 | 0080 | TSTI EQU | >80 | BTJZ test for input |
| 0273 | 0040 | TSTIU EQU | >40 | BTJO test for input or update |
| 0274 | 0000 | TSTNA EQU | >00 | BTJO test for not append |
| 0275 | 0080 | TSTOU EQU | >80 | BTJO test for output or updat |
| 0276 | * | | | |
| 0277 | | * miscellaneous constant equates | | |
| 0278 | 0020 | BLANK EQU | >20 | ASCII blank character |
| 0279 | 008C | BRDPC EQU | >8C | branch opcode |
| 0280 | 0000 | FREE EQU | >00 | unused space in table |
| 0281 | 0099 | GAPDAT EQU | >99 | non-sync data |
| 0282 | 000F | LSN EQU | >0F | mask to clear msn |
| 0283 | 00AA | SYNCDT EQU | >AA | sync data |
| 0284 | 0000 | ZERO EQU | >00 | constant |

| | | | |
|------|------------------------|-----------------------------|--------------------------|
| 0286 | * | | |
| 0287 | * command code equates | | |
| 0288 | 000E | CCATAL EQU >0E | catalog c code |
| 0289 | 000D | CFORMA EQU >0D | format c code |
| 0290 | 0001 | CCLOSE EQU >01 | close c code |
| 0291 | 0002 | CCLSDL EQU >02 | close & delete c code |
| 0292 | 0006 | CDELET EQU >06 | delete c code |
| 0293 | 00FE | CNULL EQU >FE | null c code |
| 0294 | 0000 | COPEN EQU >00 | open c code |
| 0295 | 0003 | CREAD EQU >03 | read data c code |
| 0296 | 00FF | CRESET EQU >FF | reset c code |
| 0297 | 0005 | CPOSIT EQU >05 | position record c code |
| 0298 | 0007 | CRSTAT EQU >07 | return status c code |
| 0299 | 000C | CVERIF EQU >0C | verify c code |
| 0300 | 0004 | CWRITE EQU >04 | write data c code |
| 0301 | * | | |
| 0302 | * | * command code test equates | |
| 0303 | 0008 | TCATAL EQU >08 | test for CATALOG |
| 0304 | 0001 | TFORMA EQU >01 | test for FORMAT |
| 0305 | 0008 | TNCATA EQU >08 | test for not CATALOG |
| 0306 | 0004 | TNOPEN EQU >04 | test for not OPEN |
| 0307 | 0004 | TOPEN EQU >04 | test for OPEN |
| 0308 | * | | |
| 0309 | * | * status code equates | |
| 0310 | 0013 | RSAPP EQU >13 | append mode not allowed |
| 0311 | 0002 | RSATTR EQU >02 | attribute error code |
| 0312 | 000C | RSBLEN EQU >0C | buffer length error code |
| 0313 | 001B | RSBUS EQU >1B | peripheral bus error |
| 0314 | 0001 | RSCHAR EQU >01 | device/file characterist |
| 0315 | 0004 | RSCLOS EQU >04 | no file open error code |
| 0316 | 0010 | RSDATA EQU >10 | data error |
| 0317 | 0006 | RGDEVI EQU >06 | device error code |
| 0318 | 0008 | RSLEN EQU >08 | data/file too long error |
| 0319 | 001A | RSDRCT EQU >1A | no directory error code |
| 0320 | 0007 | RSEOF EQU >07 | EOF error code |
| 0321 | 0020 | RSEOT EQU >20 | wafer full error code |
| 0322 | 0008 | RSFILE EQU >08 | too many files error cod |
| 0323 | 0003 | RSFIND EQU >03 | file not found error cod |
| 0324 | 0050 | RSLAST EQU >50 | not last file for append |
| 0325 | 0019 | RSLOWB EQU >19 | low battery warning code |
| 0326 | 000A | RSNOTI EQU >0A | it wasn't me response |
| 0327 | 0000 | RSOK EQU >00 | normal completion status |
| 0328 | 0005 | RSOPEN EQU >05 | file already open error |
| 0329 | 0009 | RSPROT EQU >09 | write protect error code |
| 0330 | 000F | RSREAD EQU >0F | read in write-only mode |
| 0331 | 0011 | RSRELA EQU >11 | relative files not suppo |
| 0332 | 000D | RSSUPP EQU >0D | unsupported command erro |
| 0333 | 0017 | RSTYPE EQU >17 | file type error |
| 0334 | 0018 | RSVERI EQU >18 | verify error code |
| 0335 | 000E | RSWRIT EQU >0E | write in read-only mode |
| 0336 | * | RSREC EQU >55 | record not found |

| | | | |
|------|--------------------|--|-------------------------|
| 0338 | * | | |
| 0339 | * trap equates | | |
| 0340 | 0017 ADDCHK EQU 23 | | add A to checksum |
| 0341 | 0008 BITEST EQU 11 | | wait end of bit count |
| 0342 | 000E BIT10A EQU 14 | | wait, bitcon=10, A->B |
| 0343 | 000F BIT4BA EQU 15 | | wait, bitcon=4, B->A |
| 0344 | 0010 BIT8AB EQU 16 | | wait, bitcon=8, A->B |
| 0345 | 0011 BIT8BA EQU 17 | | wait, bitcon=8, B->A |
| 0346 | 000D DATFIL EQU 13 | | wait, bitcon=1, CLR B |
| 0347 | 0015 DEVERR EQU 21 | | device error handler |
| 0348 | 0013 EOFTST EQU 19 | | test for EOF |
| 0349 | 000C EDTCHK EQU 12 | | test for EOT |
| 0350 | 0004 ERRWT EQU 04 | | error handler |
| 0351 | 0009 FILDAT EQU 09 | | get file data |
| 0352 | 0014 RCOMPR EQU 20 | | compare wafer/RAM data |
| 0353 | 0008 RCVCNT EQU 08 | | receive PAB/data cont. |
| 0354 | 0005 REDRUM EQU 05 | | kill open file |
| 0355 | 0007 SCREWD EQU 07 | | amp1 defaults to trap 7 |
| 0356 | 0012 TBUS EQU 18 | | transmit data over bus |
| 0357 | 0016 TFLOAT EQU 22 | | set HSK float |
| 0358 | 000A TIMEX EQU 10 | | start timer operation |
| 0359 | 0006 XMTCNT EQU 06 | | transmit response cont. |

| | | | | |
|------|------------------|------|-------------------------|------------------------------|
| 0361 | ----- | | | |
| 0362 | * origin | | | |
| 0363 | F806 | AORG | >F806 | the place where it all began |
| 0364 | ----- | | | |
| 0365 | * initialization | | | |
| 0366 | F806 | D5 | INIT CLR FFLAG | initialize no directory |
| F807 | 37 | | | |
| 0367 | F808 | A2 | EOM MOVP %I1230, IOCNTL | initialize INT1, 2&3 |
| F809 | AA | | | |
| F80A | 00 | | | |
| 0368 | F80B | 52 | MOV %STACK-1, B | R2 is BITCON |
| F80C | 02 | | | |
| 0369 | F80D | 0D | LDSP | set up stack ptr |
| 0370 | F80E | 05 | EINT | enable interrupts |
| 0371 | F80F | A2 | MOVP %INHIB, BCNTL | enable IBC at S0M |
| F810 | 04 | | | |
| F811 | 11 | | | |
| 0372 | F812 | A2 | MOVP %INITDR, DRIVE | reset drive (sensor) |
| F813 | FF | | | |
| F814 | 06 | | | |
| 0373 | F815 | A2 | MOVP %SLEEP, TIMER | set sleep mode |
| F816 | 00 | | | |
| F817 | 03 | | | |
| 0374 | F818 | A2 | MOVP %I1CS, IOCNTL | select & clear INT1 |
| F819 | 83 | | | |
| F81A | 00 | | | |
| 0375 | F81B | 01 | IDLE | wait start of message |
| 0376 | F81C | A2 | MOVP %SN, DRIVE | turn on sensors & LED |
| F81D | FB | | | |
| F81E | 06 | | | |
| 0377 | F81F | D5 | CLR COUNT | device code length -1 |
| F820 | 2F | | | |
| 0378 | F821 | 72 | MOV %SAB, DATAP | data pointer |
| F822 | 2D | | | |
| F823 | 36 | | | |
| 0379 | F824 | 8E | CALL @RCVPAB | receive device code |
| F825 | FF27 | | | |
| 0380 | F827 | B2 | DEC A | decrement device code |
| 0381 | F828 | E7 | JNC ITSME | test for device zero |
| F829 | 07 | | | |
| 0382 | F82A | 91 | MOVP TEST, B | get device address |
| F82B | 04 | | | |
| 0383 | F82C | 53 | AND %>07, B | mask off 5 ms bits |
| F82D | 07 | | | |
| 0384 | F82E | 6D | CMP B, A | compare 3 ls bits |
| 0385 | F82F | E6 | JNE EOM | test device code match |
| F830 | D7 | | | |
| 0386 | F831 | 72 | ITSME MOV %7, COUNT | rest of PAB length -1 |
| F832 | 07 | | | |
| F833 | 2F | | | |
| 0387 | F834 | F7 | TRAP RCVCNT | receive rest of PAB |
| 0388 | F835 | 32 | MOV CCODE, B | index command tables |
| F836 | 2C | | | |
| 0389 | F837 | 50 | CMP %CNULL, B | test for NULL command |
| F838 | FE | | | |
| 0390 | F839 | E2 | JEQ EOM | |
| F83A | CD | | | |

| | | | | | | |
|------|------|------|-------|-------------------|----------------------------|--------------------------|
| 0391 | F83B | 73 | AND | %WPERET, FFLAG | reset WP, error, E01 flags | |
| | F83C | 9B | | | | |
| | F83D | 37 | | | | |
| 0392 | F83E | A7 | BTJZP | %WP, WAFER, MOOSE | test WP sensor | |
| | F83F | 04 | | | | |
| | F840 | 0A | | | | |
| | F841 | 03 | | | | |
| 0393 | F842 | 74 | OR | %FWP, FFLAG | set WP flag | |
| | F843 | 20 | | | | |
| | F844 | 37 | | | | |
| 0394 | F845 | 5D | MOOSE | CMP | %RESET, B | test for RESET command |
| | F846 | FF | | | | |
| 0395 | F847 | E2 | JEQ | XRESET | | |
| | F848 | 51 | | | | |
| 0396 | F849 | 5D | CMP | %OF, B | | test for unsupported com |
| | F84A | 0F | | | | |
| 0397 | F84B | E3 | JHS | UNSUPP | | |
| | F84C | 1F | | | | |
| 0398 | F84D | CF | RLC | B | | prepare table index |
| 0399 | F84E | AA | LDA | @CTABLE(B) | | push address of command |
| | F84F | FF92 | | | | |
| 0400 | F851 | B8 | PUSH | A | | |
| 0401 | F852 | AA | LDA | @CTABLE+1(B) | | OKCRAIG |
| | F853 | FF93 | | | | |
| 0402 | F855 | B8 | PUSH | A | | |
| 0403 | F856 | 0A | RETS | | | sneaky branch to command |

| | | | | |
|-----------|-------|-------------|----------------|----------------------|
| 0405 | ----- | | | |
| 0406 F857 | D5 | RETDLO CLR | STATUS | DL = 0, OK status |
| F858 | 2D | | | |
| 0407 F859 | 72 | RETDL MOV | %1,COUNT | response length-1 |
| F85A | 01 | | | |
| F85B | 2F | | | |
| 0408 F85C | D5 | CLR | DLEN | DL = 0, status ready |
| F85D | 26 | | | |
| 0409 F85E | D5 | RETDL1 CLR | DLEN-1 | |
| F85F | 25 | | | |
| 0410 F860 | 72 | MOV | %DLEN, DATAP | data pointer |
| F861 | 26 | | | |
| F862 | 36 | | | |
| 0411 F863 | F9 | RETDL2 TRAP | XMTCNT | return zero DL |
| 0412 F864 | D5 | RTSTAT CLR | COUNT | status length-1 |
| F865 | 2F | | | |
| 0413 F866 | 72 | MOV | %STATUS, DATAP | data pointer |
| F867 | 2D | | | |
| F868 | 36 | | | |
| 0414 F869 | F9 | TRAP | XMTCNT | return status |
| 0415 F86A | E0 | EOM2 JMP | EOM | wait for EOM |
| F86B | 9C | | | |

0417 *-----
0418 * unsupported commands
0419 F86C FB UNSUPP TRAP ERRWT error
0420 F86D OD BYTE RSSUPP unsupported command code
0421 *-----
0422 * error handler nice work Craig
0423 F86E 06 WTERR DINT no uncontrolled interrupt
0424 F86F A2 MOVP %STOP, DRIVE turn off drive
F870 FB
F871 06
0425 F872 C9 POP B get pc lsb
0426 F873 B9 POP A get pc msb
0427 F874 9A LDA *B get error code
F875 01
0428 F876 D0 MOV A, STATUS store it
F877 2D
0429 * receive and discard rest of message
0430 F878 06 RCVDMO DINT no interrupt interference
0431 F879 E9 TRAP TFLOAT let HSK float
0432 F87A 72 MOV %WAKEUP/256, INT2V-1 set up INT2 vector
F87B FF
F87C 33
0433 F87D 72 MOV %WAKEUP-(256*(WAKEUP/256)), INT2V
F87E 7F
F87F 34
0434 F880 72 MOV %BROPC, INT2 set up vector branch
F881 80
F882 32
0435 F883 A2 MOVP %>80, TIME
F884 80
F885 02
0436 F886 A2 MOVP %MAX, TIMER start timer
F887 BF
F888 03
0437 F889 A2 MOVP %I2CS, IOCNTL select & clear INT2
F88A 8C
F88B 00
0438 F88C 05 EINT
0439 F88D A7 RCVDM1 BTJZP %I2S, IOCNTL, RETDL test for INT2 de-selecte
F88E 04
F88F 00
F890 C8
0440 F891 A7 BTJZP %IRQ, BSTAT, RCVDM1 test for inactive bus
F892 08
F893 11
F894 F8
0441 F895 A2 MOVP %RELEAS, BCNTL reset IRQ
F896 01
F897 11
0442 F898 E0 JMP RCVDMO
F899 DE

0444 *-----
0445 * bus reset command, all open files are closed
0446 F89A 77 XRESET BTJZ %FOPEN,FFLAG,EOM2 stop if no open file
F89B 10
F89C 37
F89D CC
0447 * CLOSE joins RESET command here
0448 F89E 73 XCLOSE AND %OPNRST,FFLAG reset OPEN flag
F89F EF
F8A0 37
0449 F8A1 77 BTJZ %TSTI,ATTRIB,XCLORS test for input
F8A2 80
F8A3 22
F8A4 0A
0450 F8A5 76 BTJO %FWP,FFLAG,XCLSD2 test for WP
F8A6 20
F8A7 37
F8A8 13
0451 F8A9 32 MOV NFILE,B current file number to B
F8AA 3E
0452 F8AB F2 TRAP DATFIL store file parameters
0453 * DELETE OPEN & DELETE may join CLOSE & RESET commands here
0454 F8AC 8E XCLWD CALL @WDIREC write directory
F8AD FC86
0455 F8AF 76 XCLORS BTJO %>80,CCODE,EOM2 test for RESET command
F8B0 80
F8B1 2C
F8B2 B7
0456 F8B3 E0 JMP RETDLO return OK status
F8B4 A2

0458 *-----
0459 * close file and delete if not write protected
0460 F8B5 73 XCLSDL AND %OPNRST,FFLAG close file
F8B6 EF
F8B7 37
0461 F8B8 77 BTJZ %FWP,FFLAG,XDELET test no write protect
F8B9 20
F8BA 37
F8BB 02
0462 F8BC FB XCLSD2 TRAP ERRWT error
0463 F8BD 09 BYTE RSPROT WP error
0464 * DELETE command joins CLOSE & DELETE here
0465 F8BE 32 XDELET MOV NFILE,B current file number to B
F8BF 3E
0466 F8C0 73 AND %INACTV,RECFIL-1 inactivate, find last
F8C1 7F
F8C2 38
0467 F8C3 F2 TRAP DATFIL
0468 F8C4 77 BTJZ %LAST,RECFIL-1,XCLOWD test for not last file
F8C5 40
F8C6 38
F8C7 E4
0469 F8C8 73 AND %LAST0,RECFIL-1 reset last file flag
F8C9 BF
F8CA 38
0470 F8CB 8E CALL @STFIL2
F8CC FEE7
0471 F8CE C1 XDELE1 TSTB test file 0
0472 F8CF E2 JZ XDELE2
F8D0 07
0473 F8D1 5A SUB %4,B decrement index (x4)
F8D2 04
0474 F8D3 AA LDA @FILE0(B)
F8D4 0040
0475 F8D6 E5 JPZ XDELE1 test for not active
F8D7 F6
0476 F8D8 24 XDELE2 OR %LAST,A set last if active or 0
F8D9 40
0477 F8DA AB STA @FILE0(B)
F8DB 0040
0478 F8DD EO JMP XCLOWD join CLOSE
F8DE CD

0480 *-----
0481 * position record
0482 F8DF 42 XPOSIT MOV NFILE, RNUM save file number
F8E0 3E
F8E1 2A
0483 F8E2 8E CALL @KILLER part of kill
F8E3 FC01
0484 F8E5 73 AND %FNUMBR, FFLAG find file by number
F8E6 FD
F8E7 37
0485 F8E8 8E CALL @POSITN
F8E9 FE2B
0486 F8EB 76 BTJD %FFOUND, FFLAG, XPOSI2 test for file found
F8EC 08
F8ED 37
F8EE 03
0487 F8EF 8E CALL @LOST2
F8F0 FF23
0488 F8F2 8C XPOSI2 BR @RETDLO return DL = 0 & status
F8F3 F857

0490 *-----
0491 * receive bus & write wafer
0492 F8F5 76 XWRITE BTJO %FWP, FFLAG, XCLSD2 test for WP
F8F6 20
F8F7 37
F8F8 C3
0493 F8F9 76 BTJO %TSTOU, ATTRIB, XWRIT2 test for output/update
F8FA 80
F8FB 22
F8FC 02
0494 F8FD FB TRAP ERRWT error
0495 F8FE 0E BYTE RSWRIT write in read only mode
0496 F8FF 77 XWRIT2 BTJZ %D80, DLEN-1, XWRIT3 test for DL too big
F900 80
F901 25
F902 02
0497 F903 FB TRAP ERRWT error
0498 F904 08 BYTE RSDLEN data too long
0499 F905 D3 XWRIT3 INC NREC increment record number
F906 3D
0500 F907 79 ADC %0, NREC-1
F908 00
F909 3C
0501 F90A 8E CALL @WSYNC write record header
F90B FBBD
0502 F90D E9 TRAP TFLOAT let HSK float
0503 F90E 8E CALL @WDL write DL
F90F FC25
0504 F911 76 BTJO %EOTFLG, FFLAG, RTDL3 test for EOT found
F912 04
F913 37
F914 1D
0505 F915 42 MOV NREC, RECFIL number of records
F916 3D
F917 39
0506 F918 73 AND %F0, RECFIL-1
F919 F0
F91A 38
0507 F91B 44 OR NREC-1, RECFIL-1
F91C 3C
F91D 38
0508 F91E 4D CMP DLEN-1, MAXLEN-1 maximum record length
F91F 25
F920 3A
0509 F921 E7 JL XWRIT4 test for MRL LT DL
F922 07
0510 F923 E6 JNE RTDL2 test for MRL GT DL
F924 08
0511 F925 4D CMP DLEN, MAXLEN
F926 26
F927 38
0512 F928 E3 JHS RTDL2 test for MRL LE DL
F929 06
0513 F92A 42 XWRIT4 MOV DLEN, MAXLEN new MRL
F92B 26
F92C 38
0514 F92D 42 MOV DLEN-1, MAXLEN-1

| | | | |
|-----------|----|--|-------------------|
| F92E | 25 | | |
| F92F | 34 | | |
| 0515 F930 | D5 | RTDL2 CLR STATUS | return OK status |
| F931 | 2D | | |
| 0516 | | * at this point status has been set to 00, 10, or 20 | |
| 0517 F932 | D5 | RTDL3 CLR DLEN-1 | rest of DL = 0 |
| F933 | 25 | | |
| 0518 F934 | C5 | CLR B | |
| 0519 F935 | 72 | MOV %1,COUNT | DL length-1 |
| F936 | 01 | | |
| F937 | 2F | | |
| 0520 F938 | 72 | MOV %DLEN, DATAP | msb DL = 0 |
| F939 | 26 | | |
| F93A | 36 | | |
| 0521 F93B | 8E | CALL @WNXPA2 | return rest of DL |
| F93C FF43 | | | |
| 0522 F93E | 8C | BR @RTSTAT | return status |
| F93F F864 | | | |

| | | | | |
|-----------|--------------------------------------|--------------|------------------|---------------------------|
| 0524 | ----- | | | |
| 0525 | * verify record | | | |
| 0526 F941 | EC | XVERIF TRAP | EOFSTST | compare nrec, recfil |
| 0527 F942 | E2 | JZ | XVERI2 | test for not EOF |
| F943 | 02 | | | |
| 0528 F944 | FB | XVERIO TRAP | ERRWT | error |
| 0529 F945 | 07 | BYTE | RSEOF | EOF error |
| 0530 F946 | 80 | XVERI2 CLRC | | clear carry |
| 0531 F947 | DF | RLC | DLEN | shift count for nibbles |
| F948 | 26 | | | |
| 0532 F949 | DF | RLC | DLEN-1 | |
| F94A | 25 | | | |
| 0533 F94B | 8E | CALL | @RSINC | inc rec#, read sync & hea |
| F94C FC9F | | | | |
| 0534 F94E | EB | TRAP | RCOMPB | compare file/record |
| 0535 F94F | EA | TRAP | DEVERR | test for device error |
| 0536 F950 | E9 | TRAP | TFLOAT | let HSK float |
| 0537 F951 | 8E | CALL | @RDL | read DL |
| F952 FD64 | | | | |
| 0538 | * compare DL to DLEN, DLEN-1 | | | |
| 0539 F954 | 4D | CMP | DLEN, NIBCON | test for same DL |
| F955 | 26 | | | |
| F956 | 2F | | | |
| 0540 F957 | E6 | JNE | XVERI4 | |
| F958 | 05 | | | |
| 0541 F959 | 4D | CMP | DLEN-1, NIBCON-1 | test for same DL |
| F95A | 25 | | | |
| F95B | 2E | | | |
| 0542 F95C | E2 | JEQ | RCOMPB | |
| F95D | 11 | | | |
| 0543 F95E | FB | XVERI4 TRAP | ERRWT | error |
| 0544 F95F | 18 | BYTE | RSVERI | verify error |
| 0545 | * read wafer and compare to bus data | | | |
| 0546 F960 | F0 | RCOMPB2 TRAP | BIT4BA | wait end of nibble |
| 0547 F961 | 23 | AND | %LSN, A | clear msn of wafer data |
| F962 | 0F | | | |
| 0548 F963 | D0 | MOV | A, TEMP2 | save data for compare |
| F964 | 2B | | | |
| 0549 | * | TRAP | ADDCHK | add wafer data to check |
| 0550 F965 | 8E | CALL | @CHKBUS | add checksum, receive bus |
| F966 FC0E | | | | |
| 0551 F968 | 1D | CMP | TEMP2, A | compare data |
| F969 | 2B | | | |
| 0552 F96A | E2 | JEQ | RCOMPB | test for match |
| F96B | 03 | | | |
| 0553 F96C | 74 | OR | %FERROR, FFLAG | store compare-error |
| F96D | 40 | | | |
| F96E | 37 | | | |
| 0554 F96F | D2 | RCOMPB DEC | NIBCON | dec nibble counter |
| F970 | 2F | | | |
| 0555 F971 | 78 | SBB | %0, NIBCON-1 | |
| F972 | 00 | | | |
| F973 | 2E | | | |
| 0556 F974 | E3 | JC | RCOMPB2 | test end of data |
| F975 | EA | | | |
| 0557 F976 | 8E | RBYTEJ CALL | @RBYTE3 | |
| F977 FDAE | | | | |

MT358F10 MLP FAMILY ASSEMBLER 1.0

12:43:22 8/ 9/82
PAGE 0020

0558 F979 E0 JMP RTDL3 return 5 more nibbles
F97A B7

0560 *-----
0561 * read wafer & transmit bus
0562 F97B 76 XREAD BTJO %TSTIU,ATTRIB,XREAD1 test for input/update
F97C 40
F97D 22
F97E 02
0563 F97F FB TRAP ERRWT error
0564 F980 0F BYTE RSREAD read in write only mode
0565 F981 EC XREAD1 TRAP EOFTST compare nrec,recfil
0566 F982 E6 JNZ XVERIO test for EOF
F983 C0
0567 * if not at EOF, implies a write was not last if in update m
0568 F984 8E CALL @RSINC inc rec#, read sync & hea
F985 FC9F
0569 F987 E8 TRAP RCOMPR compare file/record
0570 F988 EA TRAP DEVERR test for device error
0571 F989 E9 TRAP TFLOAT let HSK float
0572 F98A 8E CALL @RDLX read DL & transmit bus
F98B FD72
0573 F98D E0 JMP RNIB
F98E 05
0574 * read wafer & transmit bus (critical path 12Kbaud, 2.5 Mhz)
0575 F98F F0 RNIB2 TRAP BIT4BA wait end of nibble
0576 F990 ED TRAP TBUS transmit data over bus
0577 F991 23 AND %LSN,A clear msn for checksum
F992 0F
0578 F993 E8 TRAP ADDCHK add data to checksum
0579 F994 D2 RNIB DEC NIBCON dec nibble counter
F995 2F
0580 F996 78 SBB %0,NIBCON-1
F997 00
F998 2E
0581 F999 E3 JC RNIB2 test end of data
F99A F4
0582 F99B 8E CALL @RBYTE3
F99C FDAE
0583 F99E 32 MOV STATUS,B get status of read
F99F 2D
0584 F9A0 D5 CLR COUNT status length -1
F9A1 2F
0585 * DATAP wont be used
0586 F9A2 8E CALL @WNXPA2 return rest of status
F9A3 FF43
0587 F9A5 8C BR @EOM response
F9A6 F808

0589 *-----
0590 * return status
0591 F9A8 EC XRSTAT TRAP EOFTEST compare nrec,recfil-1
0592 F9A9 12 MOV FFLAG,A transfer flags to A
F9AA 37
0593 F9AB 23 AND %D80,A keep OPEN, WP, EOF
F9AC B0
0594 F9AD 24 OR %D07,A set seq, storage, r/w
F9AE 07
0595 F9AF D0 MOV A,BOPEN place below DLEN
F9B0 24
0596 F9B1 D5 CLR STATUS OK status
F9B2 2D
0597 F9B3 72 MOV %D01,DLEN DL = 1
F9B4 01
F9B5 26
0598 F9B6 72 MOV %D02,COUNT DL length-1
F9B7 02
F9B8 2F
0599 F9B9 8C BR @RETDL1
F9BA F85E

| | | | |
|-----------|---|----------------------------------|---------------------------|
| 0601 | *----- | | |
| 0602 | * read buffer length & attributes for OPEN only | | |
| 0603 F9BC | 7A | XCLODO SUB %3, DLEN | decrement DLEN by 3 |
| F9BD | 03 | | |
| F9BE | 26 | | |
| 0604 F9BF | E3 | JHS XCLODO | test for DL >=3 |
| F9C0 | 02 | | |
| 0605 F9C1 | FB | XCLODX TRAP ERRWT | error |
| 0606 F9C2 | 01 | BYTE RSCHAR | characteristics error |
| 0607 F9C3 | 72 | XCLODO MOV %BLOPEN, DATAP | data pointer |
| F9C4 | 24 | | |
| F9C5 | 36 | | |
| 0608 F9C6 | 72 | MOV %2, COUNT | DL of BL & attributes-1 |
| F9C7 | 02 | | |
| F9C8 | 2F | | |
| 0609 F9C9 | F7 | TRAP RCVCNT | receive BL & attributes |
| 0610 | * DELETE joins OPEN at this point | | |
| 0611 F9CA | 76 | XCLODD BTJO %>FF, DLEN-1, XCLODX | test for DL too big |
| F9CB | FF | | |
| F9CC | 25 | | |
| F9CD | F3 | | |
| 0612 F9CE | 12 | MOV DLEN, A | move DL to A |
| F9CF | 26 | | |
| 0613 F9D0 | E2 | JZ XCLODX | test for no filename |
| F9D1 | EF | | |
| 0614 F9D2 | B2 | DEC A | decrement DL (entry point |
| 0615 F9D3 | 52 | MOV %>OC, B | blank filename |
| F9D4 | 0C | | |
| 0616 F9D5 | 6D | CMP B, A | compare filename length |
| 0617 F9D6 | E3 | JHS XCLODX | test for filename too lo |
| F9D7 | E9 | | |
| 0618 F9D8 | D0 | MOV A, COUNT | move DL to count |
| F9D9 | 2F | | |
| 0619 F9DA | 22 | MOV %BLANK, A | |
| F9DB | 20 | | |
| 0620 F9DC | AB | XCLOD1 STA @FNAME1->OC(B) | store blank |
| F9DD | 0015 | | |
| 0621 F9DF | CA | DJNZ B, XCLOD1 | |
| F9E0 | FB | | |
| 0622 F9E1 | 72 | MOV %FNAME1, DATAP | data pointer |
| F9E2 | 21 | | |
| F9E3 | 36 | | |
| 0623 F9E4 | F7 | TRAP RCVCNT | receive filename |
| 0624 F9E5 | 74 | OR %FNAME, FFLAG | set filename flag |
| F9E6 | 02 | | |
| F9E7 | 37 | | |
| 0625 F9E8 | 76 | BTJO %>FO, FNAME1, XCLODF | test for file name |
| F9E9 | F0 | | |
| F9EA | 21 | | |
| F9EB | 06 | | |
| 0626 F9EC | 42 | MOV FNAME1, RNUM | mov file number to RNUM |
| F9ED | 21 | | |
| F9EE | 2A | | |
| 0627 | * CATALOG joins OPEN and DELETE at this point | | |
| 0628 F9EF | 73 | XCLODC AND %FNUMBR, FFLAG | reset file number flag |
| F9F0 | F0 | | |
| F9F1 | 37 | | |

0629 * FORMAT joins OPEN, DELETE, and CATALOG at this point
0630 F9F2 77 XCLODF BTJZ %FOPEN, FFLAG, XCLOD3 test for no file open
F9F3 10
F9F4 37
F9F5 02
0631 F9F6 FB TRAP ERRWT error
0632 F9F7 05 BYTE RSOPEN file open error
0633 F9F8 32 XCLOD3 MOV CCODE, B move ccode to B
F9F9 20
0634 F9FA 56 BTJO %TFORMA, B, XCLOD5 test for FORMAT command
F9FB 01
F9FC 3B
0635 F9FD 57 BTJZ %TNCATA, B, RDIREC test for not CATALOG
F9FE 08
F9FF 04
0636 FA00 12 MOV RNUM, A get file number
FA01 2A
0637 FA02 E6 JNZ XCLOD6 test for not 0
FA03 3A
0638 * read directory
0639 FA04 8E RDIREC CALL @FNDEOT find EOT
FA05 FE06
0640 FA07 72 MOV %DFF, NFILE file DFF
FA08 FF
FA09 3E
0641 FA0A D5 CLR NREC record 0
FA0B 3D
0642 FA0C D5 CLR NREC-1
FA0D 3C
0643 FA0E 72 MOV %D41, DLEN directory DL
FA0F 41
FA10 26
0644 FA11 22 MOV %BOTIME, A macro count
FA12 B6
0645 FA13 8E CALL @RSYNC0 read sync & file/record
FA14 FCA7
0646 FA16 EB TRAP RCOMPR compare file/record
0647 FA17 EA TRAP DEVERR test for device error
0648 FA18 72 MOV %ENDRCT, DATAP set up data pointer
FA19 7F
FA1A 36
0649 FA1B F4 TRAP BITEST wait end of nibble
0650 FA1C 72 MOV %4, BITCON restart bit count
FA1D 04
FA1E 02
0651 * 2nd half of 1st byte of RBYTE is being picked up
0652 FA1F 8E CALL @RBYTE read directory data
FA20 FDD5
0653 FA22 32 MOV CCODE, B move ccode to B
FA23 20
0654 FA24 56 BTJO %TCATAL, B, XCLOD6 test for CATALOG
FA25 08
FA26 17
0655 FA27 56 BTJO %TNOPEN, B, XCLOD5 test for OPEN command
FA28 04
FA29 0E
0656 FA2A 76 BTJO %SEQUEN, ATTRIB, XCLOD4 test for relative file

| | | | | |
|-----------|------|-------------|------------------------|----------------------------|
| FA2B | 20 | | | |
| FA2C | 22 | | | |
| FA2D | 04 | | | |
| 0657 FA2E | 76 | BTJO | %TSTNA, ATTRIB, XCLOD9 | test for not append |
| FA2F | C0 | | | |
| FA30 | 22 | | | |
| FA31 | 02 | | | |
| 0658 FA32 | FB | XCLOD4 TRAP | ERRWT | error |
| 0659 FA33 | 02 | BYTE | RSATTR | attributes byte |
| 0660 FA34 | 77 | XCLOD9 BTJZ | %TSTI, ATTRIB, XCLOD6 | test for input |
| FA35 | 80 | | | |
| FA36 | 22 | | | |
| FA37 | 06 | | | |
| 0661 FA38 | 77 | XCLOD5 BTJZ | %FWP, FFLAG, XCLOD6 | test for WP |
| FA39 | 20 | | | |
| FA3A | 37 | | | |
| FA3B | 02 | | | |
| 0662 FA3C | FB | TRAP | ERRWT | |
| 0663 FA3D | 09 | BYTE | RSPROT | write protect err |
| 0664 FA3E | 56 | XCLOD6 BTJO | %TFORMA, B, XCLOD7 | test for FORMAT command |
| FA3F | 01 | | | |
| FA40 | 0C | | | |
| 0665 FA41 | 8E | CALL | @POSITN | search for file (no error) |
| FA42 | FE2B | | | |
| 0666 FA44 | 32 | MOV | CCODE, B | move command code to B |
| FA45 | 20 | | | |
| 0667 FA46 | 57 | BTJZ | %TOPEN, B, XCLOD7 | test for OPEN command |
| FA47 | 04 | | | |
| FA48 | 04 | | | |
| 0668 FA49 | 77 | BTJZ | %FFOUND, FFLAG, XOPENO | test for file not found |
| FA4A | 08 | | | |
| FA4B | 37 | | | |
| FA4C | 1E | | | |
| 0669 FA4D | B0 | XCLOD7 CLRC | | clear carry for shift |
| 0670 FA4E | CF | RLC | B | prepare index |
| 0671 FA4F | AA | XCLOD8 LDA | @CTABLX(B) | push address of command |
| FA50 | FFB0 | | | |
| 0672 FA52 | BB | PUSH | A | |
| 0673 FA53 | AA | LDA | @CTABLX+1(B) | |
| FA54 | FFB1 | | | |
| 0674 FA56 | BB | PUSH | A | |
| 0675 FA57 | 0A | RETS | | sneaky branch |

0677 *-----
0678 * test for valid directory in ram and file open
0679 FA58 76 XOPNDR BTJO %FOPEN,FFLAG,XOPND3 test for file open
FA59 10
FA5A 37
FA5B 02
0680 FA5C FB TRAP ERRWT
0681 FA5D 04 BYTE RSCLOS file not open
0682 FA5E E0 XOPND3 JMP XCL0DB 2nd pass command decode
FA5F EF

0684 *-----
0685 * open command
0686 FA60 77 XOPEN BTJZ %FFFOUND, FFLAG, XOPNEW test for file not found
FA61 08
FA62 37
FA63 03
0687 FA64 8C BR @XOPEN4
FA65 FAF5
0688 * if file not found
0689 FA67 76 XOPNEW BTJO %TSTOU, ATTRIB, XOPNOO test for output/update
FA68 80
FA69 22
FA6A 06
0690 FA6B FB XOPENO TRAP ERRWT error
0691 FA6C 03 BYTE RSFIND file not found
0692 * attempt to OPEN in output mode
0693 FA6D 76 XOPENO BTJO %LAST, RECFIL-1, XOPNO4 to OPEN last for output
FA6E 40
FA6F 38
FA70 40
0694 * open for output or update
0695 FA71 77 XOPNOO BTJZ %LAST, FILEF, XOPNO1 test file capacity
FA72 40
FA73 7C
FA74 02
0696 FA75 FB TRAP ERRWT error
0697 FA76 08 BYTE RSFILE too many files
0698 FA77 42 XOPNO1 MOV NFILE, TEMP4 remember present file #
FA78 3E
FA79 29
0699 FA7A 8E CALL @FNDEOD find end of data
FA7B FE67
0700 FA7D 77 BTJZ %FFFOUND, FFLAG, XOPNOS test for no active files
FA7E 08
FA7F 37
FA80 02
0701 FA81 D3 INC NFILE next file number
FA82 3E
0702 * write file header (if EOT, no return)
0703 FA83 72 XOPNOS MOV %D0C, DLEN filename length
FA84 0C
FA85 26
0704 FA86 D5 CLR NREC file header record #
FA87 3D
0705 FA88 D5 CLR NREC-1
FA89 3C
0706 FA8A 8E CALL @WSYNC write record header
FA8B FB8D
0707 FA8D 72 MOV %FNAME1, DATAP data pointer
FA8E 21
FA8F 36
0708 FA90 8E CALL @WMEM write filename & checksum
FA91 FC99
0709 * reset old file active/last flag if necessary
0710 FA93 32 MOV TEMP4, B recall past file
FA94 29
0711 FA95 F6 TRAP FILDAT recall file data

| | | | | | |
|------|------|------|-------------|---|-------------------------|
| 0712 | FA96 | 77 | BTJZ | %LAST, RECFIL-1, XOPN02 | test for not last file |
| | FA97 | 40 | | | |
| | FA98 | 38 | | | |
| | FA99 | 05 | | | |
| 0713 | FA9A | 73 | AND | %LAST0, RECFIL-1 | reset last flag |
| | FA9B | BF | | | |
| | FA9C | 38 | | | |
| 0714 | FA9D | E0 | JMP | XOPN03 | |
| | FA9E | 03 | | | |
| 0715 | FA9F | 73 | XOPN02 AND | %INACTV, RECFIL-1 | reset active flag |
| | FAA0 | 7F | | | |
| | FAA1 | 38 | | | |
| 0716 | FAA2 | 8E | XOPN03 CALL | @STFIL2 | save past file |
| | FAA3 | FEE7 | | | |
| 0717 | FAA5 | 32 | MOV | NFILE, B | get current file number |
| | FAA6 | 3E | | | |
| 0718 | FAA7 | C2 | DEC | B | get previous file # |
| 0719 | FAA8 | E7 | JNC | XOPN04 | test for B not >FF |
| | FAA9 | 07 | | | |
| 0720 | FAAA | F6 | TRAP | FILDAT | get file data |
| 0721 | FAAB | 73 | AND | %LAST0, RECFIL-1 | reset last flag |
| | FAAC | BF | | | |
| | FAAD | 38 | | | |
| 0722 | FAAE | 8E | CALL | @STFIL2 | put file data |
| | FAAF | FEE7 | | | |
| 0723 | | | | * OPEN existing last file for output joins here | |
| 0724 | FAB1 | 72 | XOPN04 MOV | %NEWFIL, RECFIL-1 | new file (active/last) |
| | FAB2 | CO | | | |
| | FAB3 | 38 | | | |
| 0725 | FAB4 | DS | CLR | RECFIL | |
| | FAB5 | 39 | | | |
| 0726 | FAB6 | D5 | CLR | MAXLEN-1 | |
| | FAB7 | 3A | | | |
| 0727 | FAB8 | D5 | CLR | MAXLEN | |
| | FAB9 | 3B | | | |
| 0728 | FABA | 77 | BTJZ | %INTDIS, ATTRIB, XOPEN1 | test for display format |
| | FABB | 08 | | | |
| | FABC | 22 | | | |
| | FABD | 03 | | | |
| 0729 | FABE | 74 | OR | %INTERN, RECFIL-1 | set internal file type |
| | FABF | 10 | | | |
| | FAC0 | 38 | | | |
| 0730 | FAC1 | 12 | XOPEN1 MOV | BOPEN-1, A | test for BL > 0 |
| | FAC2 | 23 | | | |
| 0731 | FAC3 | E2 | JZ | XOPEN5 | |
| | FAC4 | 04 | | | |
| 0732 | FAC5 | E5 | JPZ | XOPEN2 | test for DL error |
| | FAC6 | 08 | | | |
| 0733 | FAC7 | FB | TRAP | ERRWT | error |
| 0734 | FAC8 | 08 | BYTE | RSDLEN | DL too big |
| 0735 | FAC9 | 12 | XOPEN5 MOV | BOPEN, A | test for BL > 0 |
| | FACA | 24 | | | |
| 0736 | FACB | E6 | JNZ | XOPEN2 | |
| | FACC | 02 | | | |
| 0737 | FACD | D3 | INC | BOPEN-1 | 256 -> BL |
| | FACE | 23 | | | |
| 0738 | FACF | D5 | XOPEN2 CLR | STATUS | |

| | | | | | |
|------|------|------|--------|-----------------------------|--------------------------|
| | FAD0 | 2D | | | |
| 0739 | FAD1 | EC | TRAP | EOFTEST | test for EOF, set flag |
| 0740 | FAD2 | 72 | XOPEN3 | MOV %4, DLEN | prepare to send DL |
| | FAD3 | 04 | | | |
| | FAD4 | 26 | | | |
| 0741 | FAD5 | D5 | CLR | DLEN-1 | |
| | FAD6 | 25 | | | |
| 0742 | FAD7 | 72 | MOV | %1, COUNT | DL length-1 |
| | FAD8 | 01 | | | |
| | FAD9 | 2F | | | |
| 0743 | FADA | 72 | MOV | %DLEN, DATAP | |
| | FADB | 26 | | | |
| | FADC | 36 | | | |
| 0744 | FADD | F9 | TRAP | XMTCNT | return DL |
| 0745 | FADE | 72 | MOV | %1, COUNT | BL length -1 |
| | FADE | 01 | | | |
| | FAE0 | 2F | | | |
| 0746 | FAE1 | 72 | MOV | %BLOPEN, DATAP | (ms byte already 0) |
| | FAE2 | 24 | | | |
| | FAE3 | 36 | | | |
| 0747 | FAE4 | F9 | TRAP | XMTCNT | return BL |
| 0748 | FAE5 | 12 | MOV | STATUS, A | get value of status |
| | FAE6 | 2D | | | |
| 0749 | FAE7 | E6 | JNZ | XNOPEN | test to set open flag |
| | FAE8 | 03 | | | |
| 0750 | FAE9 | 72 | MOV | %NEWFLG, FFLAG | set up open flags |
| | FAEA | 14 | | | |
| | FAEB | 37 | | | |
| 0751 | FAEC | 72 | XNOPEN | MOV %1, COUNT | RN length -1 |
| | FAED | 01 | | | |
| | FAEE | 2F | | | |
| 0752 | FAEF | 72 | MOV | %NREC, DATAP | (ms byte already 0) |
| | FAF0 | 3D | | | |
| | FAF1 | 36 | | | |
| 0753 | FAF2 | 8C | BR | @RETDL2 | return BL & status |
| | FAF3 | F863 | | | |
| 0754 | | | | | * if file found |
| 0755 | FAF5 | 76 | XOPEN4 | BTJO %TSTIU, ATTRIB, XOPEN9 | test for input/update |
| | FAF6 | 40 | | | |
| | FAF7 | 22 | | | |
| | FAF8 | 03 | | | |
| 0756 | FAF9 | 8C | BR | @XOPENO | open for output |
| | FAFA | FA6D | | | |
| 0757 | FAFC | 12 | XOPEN9 | MOV ATTRIB, A | move attributes to A |
| | FAFD | 22 | | | |
| 0758 | FAFE | BE | RL | A | shift file type to bit 4 |
| 0759 | FAFF | 15 | XOR | RECFIL-1, A | compare file type |
| | FB00 | 38 | | | |
| 0760 | FB01 | 27 | BTJZ | %DISPLAY, A, XOPEN8 | test for same value |
| | FB02 | 10 | | | |
| | FB03 | 02 | | | |
| 0761 | FB04 | FB | TRAP | ERRWT | error |
| 0762 | FB05 | 17 | BYTE | RSTYPE | file type |
| 0763 | FB06 | 77 | XOPEN8 | BTJZ %TSTIU, ATTRIB, XOPEN7 | test for input |
| | FB07 | 80 | | | |
| | FB08 | 22 | | | |
| | FB09 | 06 | | | |

0764 * if update
0765 FB0A 76 BTJO %LAST, RECFIL-1, XOPEN7 test for last file
FB0B 40
FB0C 38
FB0D 02
0766 FB0E FB TRAP ERRWT error
0767 FB0F 50 BYTE RSLAST not last file for append
0768 FB10 76 XOPEN7 BTJO %OFF, BOPEN-1, XOPNIO test for BL > 0
FB11 FF
FB12 23
FB13 0C
0769 FB14 76 BTJO %OFF, BOPEN, XOPNIO test for BL > 0
FB15 FF
FB16 24
FB17 08
0770 FB18 42 MOV MAXLEN, BOPEN return MRL
FB19 3B
FB1A 24
0771 FB1B 42 MOV MAXLEN-1, BOPEN-1
FB1C 3A
FB1D 23
0772 FB1E E0 JMP XOPEN1 go test for BL = 0
FB1F A1
0773 FB20 4D XOPNIO CMP BOPEN-1, MAXLEN-1
FB21 23
FB22 3A
0774 FB23 E7 JL XOPEN2 test for MRL LT BL
FB24 AA
0775 FB25 E6 JNE XOPNII test for MRL GT BL
FB26 05
0776 FB27 4D CMP MAXLEN, BOPEN
FB28 3B
FB29 24
0777 FB2A E3 JHS XOPEN2 test for MRL GE BL
FB2B A3
0778 FB2C 72 XOPNII MOV %RSBLEN, STATUS
FB2D 0C
FB2E 2D
0779 FB2F 42 MOV MAXLEN, BOPEN return MRL with error
FB30 3B
FB31 24
0780 FB32 42 MOV MAXLEN-1, BOPEN-1
FB33 3A
FB34 23
0781 FB35 E0 JMP XOPEN3
FB36 9B

0783 *-----
0784 * delete file command is a subset of CLOSE & DELETE
0785 *-----
0786 * format wafer
0787 FB37 B5 XFORMA CLR A initialize directory
0788 FB38 52 MOV %D41,B set up index
FB39 41
0789 FB3A AB XFORM1 STA @D3E(B) no last/active files, REV
FB3B 003E
0790 FB3D CA DJNZ B, XFORM1
FB3E FB
0791 FB3F 72 MOV %DFORMA, FILE0 inactive/last
FB40 40
FB41 40
0792 FB42 8E CALL @WDIREC write directory
FB43 FC86
0793 * add format pattern write and read if there is room
0794 FB45 8C BR @RETDLO return OK status, DL = 0
FB46 FB57

0796 *-----
0797 * catalog command, return current file parameters
0798 FB48 8E XCATAL CALL @FNDEOF find EOF
FB49 FE56
0799 FB4B 76 BTJO %OFF, BLEN-1, XCATA2 test for adequate BLEN
FB4C FF
FB4D 27
FB4E 07
0800 FB4F 7D CMP %18, BLEN
FB50 12
FB51 28
0801 FB52 E3 JHS XCATA2
FB53 02
0802 FB54 FB TRAP ERRWT
0803 FB55 0C BYTE RSBLLEN BLEN error
0804 FB56 72 XCATA2 MOV %18, DLEN prepare response DL
FB57 12
FB58 26
0805 FB59 D5 CLR DLEN-1
FB5A 25
0806 FB5B 72 MOV %DLEN, DATAP set up DL pointer
FB5C 26
FB5D 36
0807 FB5E 72 MOV %1, COUNT length of DL-1
FB5F 01
FB60 2F
0808 FB61 F9 TRAP XMTCNT return DL
0809 * check on the order in which data appears in data buffer
0810 FB62 72 MOV %NFILE, DATAP prepare to send file num
FB63 3E
FB64 36
0811 FB65 D5 CLR COUNT file number length-1
FB66 2F
0812 FB67 F9 TRAP XMTCNT return file number
0813 FB68 72 MOV %FNAME1, DATAP prepare to send file nam
FB69 21
FB6A 36
0814 FB6B 72 MOV %NMLEN-1, COUNT file name length-1
FB6C 08
FB6D 2F
0815 FB6E F9 TRAP XMTCNT return file name
0816 FB6F 42 MOV RECFIL-1, TEMP1 prepare to send file sta
FB70 38
FB71 2C
0817 FB72 73 AND %LSN, RECFIL-1
FB73 0F
FB74 38
0818 FB75 72 MOV %MAXLEN, DATAP
FB76 38
FB77 36
0819 FB78 72 MOV %3, COUNT file status length-1
FB79 03
FB7A 2F
0820 FB7B F9 TRAP XMTCNT return number of records
0821 FB7C 42 MOV TEMP1, RECFIL-1

| | | | | | |
|------|------|------|------|---------------|-----------------------|
| 0822 | FB7F | 73 | AND | %DFO, TEMP1 | prepare to send flags |
| | FB80 | F0 | | | |
| | FB81 | 2C | | | |
| 0823 | FB82 | D5 | CLR | COUNT | flags length-1 |
| | FB83 | 2F | | | |
| 0824 | FB84 | 72 | MOV | %TEMP1, DATAP | |
| | FB85 | 2C | | | |
| | FB86 | 36 | | | |
| 0825 | FB87 | F9 | TRAP | XMTCNT | return flags |
| 0826 | FB88 | D5 | CLR | STATUS | store OK status |
| | FB89 | 2D | | | |
| 0827 | FB8A | 8C | BR | ERTSTAT | return status |
| | FB8B | F864 | | | |

0829 *-----
0830 * write sync pattern
0831 * XWRITE and OPEN enter here
0832 FB8D F5 WSYNC TRAP TIMEX start motor & timer
0833 FB8E A2 MOVP %SN, DRIVE stop motor
FB8F FB
FB90 06
0834 * WDIREC enters here
0835 FB91 72 WSYNC0 MOV %D01, NIBCON-1 256 nibbles of delay
FB92 01
FB93 2E
0836 FB94 8E CALL @WSYNC5 call delay routine
FB95 FBE8
0837 FB97 72 MOV %D04, NIBCON-1 1279 nibbles of gap
FB98 04
FB99 2E
0838 * (256 ms minimum delay inherent)
0839 FB9A 12 MOV NFILE,A combine file/rec bits
FB9B 3E
0840 FB9C 14 OR NREC,A
FB9D 3D
0841 FB9E 14 OR NREC-1,A
FB9F 3C
0842 FBA0 E6 JNZ WSYNC2 test for file/rec NE 0
FBA1 02
0843 FBA2 DE RL NIBCON-1 extra length gap (0,0)
FBA3 2E
0844 * 1024 nibbles of sync (512 ms delay inherent)
0845 FBA4 22 WSYNC2 MOV %GAPDAT,A non-sync data
FBA5 99
0846 FBA6 A2 MOVP %MT, DRIVE turn on MT
FBA7 F3
FBA8 06
0847 FBA9 76 BTJO %FWP, FFLAG, WGHOST test for WP
FBAA 20
FBAB 37
FBAC 06
0848 FBAD A2 MOVP %MTWE, DRIVE turn on WE
FBAE F1
FBAF 06
0849 FBB0 A2 MOVP %INITWF, DDRD set up DO to output
FBB1 01
FBB2 0B
0850 FBB3 8E WGHOST CALL @WSYNC3 write gap data
FBB4 FBD6
0851 FBB6 D5 CLR NIBCON-1 255 nibbles of sync
FBB7 2E
0852 FBB8 22 MOV %SYNCDT,A sync data
FBB9 AA
0853 FBB0 8E CALL @WSYNC3 write sync
FBBB FBD6
0854 FBB0 B5 CLR A clear A (start nibble/sp
0855 * start nibble and spacer will be written in WBYTE
0856 FB8E 72 MOV %D03, COUNT file/record length
FB8F 03
FB80 2F

0858 *-----
0859 * load and write wafer
0860 FBC1 F1 WBYTE TRAP BIT10A wait end of byte
0861 * first time : writes preset value of A (start/spacer)
0862 FBC2 9A LDA *DATAP load data
FBC3 36
0863 FBC4 D2 DEC DATAP dec data pointer
FBC5 36
0864 FBC6 E8 TRAP ADDCHK add to checksum (byte)
0865 FBC7 F3 TRAP EOTCHK test for EOT
0866 FBC8 DA DJNZ COUNT, WBYTE dec byte counter
FBC9 2F
FBCA F6
0867 FBCB F1 TRAP BIT10A wait end of byte
0868 * start writing last data byte, spacer started elsewhere
0869 FBCC 42 MOV DLEN, NIBCON move DL to nibcon
FBCD 26
FBCE 2F
0870 FBCF 42 MOV DLEN-1, NIBCON-1
FBDO 25
FBD1 2E
0871 FBD2 0A RETS

| | | | | | |
|-----------|--|--------|----------|-----------------------|-------------------------|
| 0873 | ----- | | | | |
| 0874 | | *WEOT | BR | @WEOT2 | EOT found |
| 0875 | ----- | | | | |
| 0876 FBD3 | FA | WHELP | TRAP | REDRUM | message terminated/slow |
| 0877 FBD4 | FB | | TRAP | ERRWT | error |
| 0878 FBD5 | 1B | | BYTE | RSBUS | bus malfunction |
| 0879 | ----- | | | | |
| 0880 FBD6 | EF | WSYNC3 | TRAP | BITBAB | wait, bitcon=8, A->B |
| 0881 FBD7 | F3 | | TRAP | EOTCHK | test for EOT |
| 0882 FBD8 | 76 | | BTJO | %FO, NFILE, WSYNC4 | test for WDIREC |
| FBDF9 | F0 | | | | |
| FBDA | 3E | | | | |
| FBDB | 04 | | | | |
| 0883 FBDC | A7 | | BTJZP | %BATTRY, TEST, IMDEAD | test for low battery |
| FBDD | 40 | | | | |
| FBDE | 04 | | | | |
| FBDF | 28 | | | | |
| 0884 FBE0 | D2 | WSYNC4 | DEC | NIBCON | dec nibble counter |
| FBE1 | 2F | | | | |
| 0885 FBE2 | 7B | | SBB | %0, NIBCON-1 | |
| FBE3 | 00 | | | | |
| FBE4 | 2E | | | | |
| 0886 FBE5 | E3 | | JC | WSYNC5 | test for end data |
| FBE6 | EF | | | | |
| 0887 FBE7 | 0A | | RETS | | |
| 0888 | ----- | | | | |
| 0889 | * delay routine prior to looking for EOT or writing sync | | | | |
| 0890 FBE8 | 72 | WSYNC5 | MOV | %NFILE, DATAP | file & record numbers |
| FBE9 | 3E | | | | |
| FBEA | 36 | | | | |
| 0891 FBEB | D5 | CLR | CHKSUM-1 | | initialize checksum |
| FBEC | 30 | | | | |
| 0892 FBED | D5 | CLR | CHKSUM | | |
| FBEE | 31 | | | | |
| 0893 FBEF | D5 | CLR | NIBCON | | clear 1s byte of NIBCON |
| FBF0 | 2F | | | | |
| 0894 FBF1 | EF | WSYNC6 | TRAP | BITBAB | wait, bitcon=8, A->B |
| 0895 FBF2 | D2 | | DEC | NIBCON | dec nibble counter |
| FBF3 | 2F | | | | |
| 0896 FBF4 | 7B | | SBB | %0, NIBCON-1 | |
| FBF5 | 00 | | | | |
| FBF6 | 2E | | | | |
| 0897 FBF7 | E3 | | JC | WSYNC6 | test for end data |
| FBF8 | F8 | | | | |
| 0898 FBF9 | 0A | | RETS | | |

| | | | | | |
|-----------|--|--------|--------------------------|-----------------------|-------------|
| 0900 | ----- | | | | |
| 0901 | * kill open file and rewrite directory | | | | |
| 0902 FBFA | 77 | KILL | BTJZ %FOPEN,FFLAG,KILLED | test for no open file | |
| FBFB | 10 | | | | |
| FBFC | 37 | | | | |
| FBFD | 09 | | | | |
| 0903 FBFE | 73 | AND | %OPNRST,FFLAG | close file | |
| FBFF | EF | | | | |
| FC00 | 37 | | | | |
| 0904 | * BTJO %FWP,FFLAG,KILLED | | | | test for WP |
| 0905 FC01 | 32 | KILLER | MOV NFILE,B | | |
| FC02 | 3E | | | | |
| 0906 FC03 | F2 | TRAP | DATFIL | | |
| 0907 FC04 | 8E | CALL | @WDIREC | | |
| FC05 | FC86 | | | | |
| 0908 FC07 | 0A | KILLED | RETS | | |
| 0909 | ----- | | | | |
| 0910 FC08 | 52 | IMDEAD | MOV %STACK-1,B | low battery | |
| FC09 | 02 | | | | |
| 0911 FC0A | 0D | LDSP | | clear stack | |
| 0912 FC0B | FA | TRAP | REDRUM | shut down | |
| 0913 FC0C | FB | TRAP | ERRWT | error | |
| 0914 FC0D | 19 | BYTE | RSLOWB | low battery | |

0916 *-----
0917 * receive bus data and add to checksum
0918 FCOE 48 CHKBUS ADD A, CHKSUM operant frequation
FC0F 00
FC10 31
0919 FC11 79 ADC %0, CHKSUM-1
FC12 00
FC13 30
0920 FC14 A7 RBUS BTJZP %BAV, BSTAT, WHELP test for BAV active
FC15 02
FC16 11
FC17 BB
0921 FC18 A7 BTJZP %IRQ, BSTAT, WHELP test for bus data ready
FC19 08
FC1A 11
FC1B B7
0922 FC1C A2 MOVP %RELEASE, BCNTL reset IRQ
FC1D 01
FC1E 11
0923 FC1F 80 MOVP BDATA, A read data from bus
FC20 10
0924 FC21 E9 TRAP TFLOAT
0925 FC22 23 AND %LSN, A let HSK float
FC23 0F clear ms nibble of A
0926 FC24 0A RETS

0928 *-----
0929 FC25 F1 WDL TRAP BIT10A wait, bitcon=10, A->B
0930 * writes byte spacer to be picked up at end of RBYTE
0931 FC26 48 ADD DLEN,CHKSUM add to checksum
FC27 26
FC28 31
0932 FC29 79 ADC %0,CHKSUM-1
FC2A 00
FC2B 30
0933 FC2C 48 ADD DLEN-1,CHKSUM
FC2D 25
FC2E 31
0934 FC2F 79 ADC %0,CHKSUM-1
FC30 00
FC31 30
0935 FC32 12 MOV DLEN,A load ls byte of DI
FC33 26
0936 FC34 F1 TRAP BIT10A wait, bitcon=10, A->B
0937 FC35 12 MOV DLEN-1,A load ms byte of DI
FC36 25
0938 FC37 F1 TRAP BIT10A wait, bitcon=10, A->B
0939 FC38 78 ADD %>08,BITCON extra nibble for spacer
FC39 08
FC3A 02
0940 * 2nd spacer will be written in WNIB

0942 *-----
0943 * receive bus & write wafer
0944 FC3B DF WNIB RLC NIBCON multiply nibcon by 2
FC3C 2F
0945 FC3D DF RLC NIBCON-1 (byte count -> nibble co
FC3E 2E
0946 FC3F E0 JMP WNIB3
FC40 06
0947 FC41 EF WNIB0 TRAP BITBAB wait end of nibble
0948 * first time : writes 2nd spacer after DL
0949 FC42 F3 TRAP EOTCHK test for EOT
0950 FC43 8E CALL @RBUS receive bus data
FC44 FC14
0951 FC46 E8 TRAP ADDCHK add to checksum
0952 FC47 D2 WNIB3 DEC NIBCON dec nibble counter
FC48 2F
0953 FC49 7B SBB %0, NIBCON-1
FC4A 00
FC4B 2E
0954 FC4C E3 JC WNIB0 test for end data
FC4D F3
0955 * last data byte and spacer will be written in WCHKSM

0957 *-----
0958 * write checksum, turn off drive and wait for stop
0959 FC4E F1 WCHKSM TRAP BIT10A wait, bitcon=10, A->B
0960 * first time : writes preset value of A (bus datat/spacer)
0961 FC4F 12 MOV CHKSUM,A load 1s byte of checksum
FC50 31
0962 FC51 F1 TRAP BIT10A wait, bitcon=10, A->B
0963 FC52 12 MOV CHKSUM-1,A load ms byte of checksum
FC53 30
0964 FC54 F1 TRAP BIT10A wait, bitcon=10, A->B
0965 FC55 DF RLC BITCON extra byte
FC56 02
0966 * write out last checksum byte & extra data for RCHKSM
0967 FC57 7D CMP %CWRITE,CCODE test for XWRITE
FC58 04
FC59 20
0968 FC5A E6 JNE WLASTO and
FC5B 06
0969 FC5C A2 MOVP %D00,BDATA begin to return DL
FC5D 00
FC5E 10
0970 FC5F A2 MOVP %DROP,BCNTL
FC60 01
FC61 11

0972 *-----
0973 * wait end of extra data so RCHKSM wont hang up
0974 FC62 F4 WLASTO TRAP BITEST wait for end
0975 * turn off drive & wait for stop, keep looking for EOT
0976 FC63 80 MOVP DRIVE,A get drive control
FC64 06
0977 FC65 25 XOR %MTBAR,A motor off
FC66 08
0978 FC67 82 MOVP A, DRIVE set drive control
FC68 06
0979 FC69 22 MOV %GAPDAT,A non-sync data
FC6A 99
0980 FC6B 8E CALL @RLASTO wait for motor to stop
FC6C FDE9
0981 FC6E 76 BTJO %EOTFLG,FFLAG,WEOT3 test for EOT found
FC6F 04
FC70 37
FC71 08
0982 FC72 0A RETS
0983 *-----
0984 * test for EOT, if found kill operation and close file
0985 FC73 A7 TSTEOT BTJZP %EOTTST,WAFER,WEOT2 test for EOT
FC74 02
FC75 0A
FC76 01
0986 FC77 0A RETS
0987 FC78 D5 WEOT2 CLR CCODE this case ignores XWRITE
FC79 2C
0988 FC7A FA WEOT3 TRAP REDRUM
0989 FC7B 7D CMP %CWRITE,CCODE shut down
FC7C 04 test for XWRITE
FC7D 2C
0990 FC7E E2 JEQ WEOT4
FC7F 02
0991 FC80 FB TRAP ERRWT error
0992 FC81 20 BYTE RSEOT wafer full
0993 FC82 72 WEOT4 MOV %RSEOT,STATUS set EOT status
FC83 20
FC84 2D
0994 FC85 0A RETS

| | | | | |
|------|------|---------------------------|---------------------|------------------------|
| 0996 | | * | | |
| 0997 | | * write directory | | |
| 0998 | FC86 | 8E | WDIREC CALL @FNDEOT | find EOT |
| | FC87 | FE06 | | |
| 0999 | | * (640 ms delay inherent) | | |
| 1000 | FC89 | 72 | MOV %FF, NFILE | file >FF |
| | FC8A | FF | | |
| | FC8B | 3E | | |
| 1001 | FC8C | D5 | CLR NREC | record 0 |
| | FC8D | 3D | | |
| 1002 | FC8E | D5 | CLR NREC-1 | |
| | FC8F | 3C | | |
| 1003 | FC90 | 72 | MOV %41, DLEN | directory DL |
| | FC91 | 41 | | |
| | FC92 | 26 | | |
| 1004 | FC93 | 8E | CALL @WSYNC0 | write sync pattern |
| | FC94 | FB91 | | |
| 1005 | FC96 | 72 | MOV %ENDRCT, DATAP | directory data pointer |
| | FC97 | 7F | | |
| | FC98 | 36 | | |
| 1006 | FC99 | 8E | WMEM CALL @WBYTE | write RAM data |
| | FC9A | FBC1 | | |
| 1007 | FC9C | 8C | BR @WCHKSM | write checksum |
| | FC9D | FC4E | | |

1009 *-----
1010 * read sync pattern
1011 * blank tape noise could be high and/or low frequency,
1012 * short and/or long bit times may be read if noise
1013 * before sync is read as sync
1014 FC9F D3 RSINC INC NREC . increment record number
FCA0 3D
1015 FCA1 79 ADC %0, NREC-1
FCA2 00
FCA3 3C
1016 FCA4 F5 RSYNC TRAP TIMEX start motor & timer
1017 FCA5 22 MOV %BOSTIM, A macro count (225 ms)
FCA6 67
1018 FCA7 A2 RSYNCO MOVP %MAX, TIMER start timer
FCA8 BF
FCA9 03
1019 FCAA D0 MOV A, BITCON move delay
FCAB 02
1020 FCAC A2 MOVP %RENITW, DDRD tri-state DO
FCAD 00
FCAE 0B
1021 FCAF F4 TRAP BITEST wait for safe area
1022 * add low battery test, stop reading and execute RESET
1023 FCB0 A6 BTJOP %BATTRY, TEST, RESYNC test for not low battery
FCB1 40
FCB2 04
FCB3 03
1024 FCB4 8C BR @IMDEAD
FCB5 FC08
1025 FCB7 06 RESYNC DINT no interrupference
1026 FCB8 72 MOV %ISYNC2/256, INT2V-1 INT2 vector
FCB9 FF
FCBA 33
1027 FCBB 72 MOV %ISYNC2-(256*(ISYNC2/256)), INT2V
FCBC 85
FCBD 34
1028 FCBE 72 MOV %BROPC, INT2
FCBF 8C
FCC0 32
1029 FCC1 A2 MOVP %OFF, TIME set up max time
FCC2 FF
FCC3 02
1030 FCC4 A2 MOVP %I23CS, IOCTL clear & select INT2&3
FCC5 BC
FCC6 00
1031 FCC7 05 EINT reinterruptidity
1032 FCC8 72 RSYNC2 MOV %>16, BITCON valid-bit counter
FCC9 16
FCCA 02
1033 FCCB C5 CLR B clear previous count
1034 FCCC D5 CLR CHKSUM clear checksum
FCCD 31
1035 FCCE D5 CLR CHKSUM-1
FCCF 30
1036 FCD0 A6 RSYNC3 BTJOP %EOTST, WAFER, RSYNXT if not EOT
FCD1 02
FCD2 04

| | | | | | |
|------|------|----|--------|---|--------------------------|
| | FCD3 | 02 | | | |
| 1037 | FCD4 | FB | TRAP | ERRWT | error |
| 1038 | FCD5 | 1A | BYTE | RSDRCT | directory error |
| 1039 | FCD6 | 01 | RSYNXT | IDLE | wait transition |
| 1040 | FCD7 | 80 | MOVP | CAPTUR, A | read bit time |
| | FCD8 | 03 | | | |
| 1041 | FCD9 | B4 | INV | A | get true count |
| 1042 | FCDA | 28 | ADD | %2, A | even more true |
| | FCDB | 02 | | | |
| 1043 | FCDC | 26 | BTJO | %HRANGE, A, RSYNC2 | data rate too slow |
| | FCDD | C0 | | | |
| | FCDE | E9 | | | |
| 1044 | FCDF | 26 | BTJO | %LRANGE, A, RSYNCA | data rate OK |
| | FCE0 | 38 | | | |
| | FCE1 | 02 | | | |
| 1045 | FCE2 | E0 | JMP | RSYNC2 | data rate too fast |
| | FCE3 | E4 | | | |
| 1046 | FCE4 | C1 | RSYNCA | TSTB | B status |
| 1047 | FCE5 | E2 | JZ | RSYNC4 | test for previous count |
| | FCE6 | 04 | | | |
| 1048 | FCE7 | 3A | SUB | A, B | compare to new count |
| | FCE8 | 00 | | | |
| 1049 | FCE9 | E3 | JC | RSYNC2 | test for difference of 4 |
| | FCEA | DD | | | |
| 1050 | FCEB | C0 | RSYNC4 | MOV | save new count |
| 1051 | FCEC | 5A | SUB | %ORANGE, B | offset for compare |
| | FCED | 06 | | | |
| 1052 | FCEE | E8 | TRAP | ADDCHK | add to cumulative |
| 1053 | FCEF | DA | DJNZ | BITCON, RSYNC3 | test end sync sample |
| | FCF0 | 02 | | | |
| | FCF1 | DE | | | |
| 1054 | | | * | test for EOT | |
| 1055 | | | * | add check for bit time in above loop (length consistency) | |
| 1056 | FCF2 | 06 | DINT | | no interrupt preference |
| 1057 | FCF3 | 72 | MOV | %5, BITCON | cumulative/ >20 |
| | FCF4 | 05 | | | |
| | FCF5 | 02 | | | |
| 1058 | FCF6 | 76 | BTJO | %>OF, CHKSUM, RSYNC5 | test rounding |
| | FCF7 | 0F | | | |
| | FCF8 | 31 | | | |
| | FCF9 | 02 | | | |
| 1059 | FCFA | E0 | JMP | RSYNC6 | no round |
| | FCFB | 06 | | | |
| 1060 | FCFC | 78 | RSYNC5 | ADD | round |
| | FCFD | 10 | | | |
| | FCFE | 31 | | | |
| 1061 | FCFF | 79 | ADC | %0, CHKSUM-1 | |
| | FD00 | 00 | | | |
| | FD01 | 30 | | | |
| 1062 | FD02 | DD | RSYNC6 | RRD | shift msb right |
| | FD03 | 30 | | | |
| 1063 | FD04 | DD | RRD | CHKSUM | |
| | FD05 | 31 | | | |
| 1064 | FD06 | DA | DJNZ | BITCON, RSYNC6 | test end of divide |
| | FD07 | 02 | | | |
| | FD08 | F9 | | | |
| 1065 | FD09 | 32 | MOV | CHKSUM, B | |

| | | | | | |
|------|------|----|--------------------|----------------------------|--------------------------|
| | FDOA | 31 | | | |
| 1066 | FD0B | 5A | SUB | %2, B | adjust bitime value |
| | FDOC | 02 | | | |
| 1067 | FD0D | 92 | MOVP | B, TIME | 2/3 bit timer |
| | FDOE | 02 | | | |
| 1068 | FD0F | D5 | CLR | CHKSUM | set chksum to 0 |
| | FD10 | 31 | | | |
| 1069 | FD11 | D5 | CLR | CHKSUM-1 | |
| | FD12 | 30 | | | |
| 1070 | * | | COPY | ALC, STRINGY, SRC, OLDRINT | old microcode int. |
| 1071 | FD13 | 72 | MOV | %RDBIT1, INT2 | set up INT2 opcode |
| | FD14 | A4 | | | |
| | FD15 | 32 | | | |
| 1072 | FD16 | 72 | MOV | %RDBIT2, INT2+1 | set up INT2 parameter |
| | FD17 | 20 | | | |
| | FD18 | 33 | | | |
| 1073 | FD19 | D5 | CLR | INT2+2 | set up INT2 parameter |
| | FD1A | 34 | | | |
| 1074 | FD1B | A2 | MOVP | %START, TIMER | start 2/3 bit timer |
| | FD1C | A0 | | | |
| | FD1D | 03 | | | |
| 1075 | FD1E | A2 | MOVP | %I2CS, IOCNTL | select & clear INT2 |
| | FD1F | 8C | | | |
| | FD20 | 00 | | | |
| 1076 | FD21 | 05 | EINT | | reinterruptidity |
| 1077 | FD22 | C5 | CLR | B | TEST |
| 1078 | FD23 | 01 | IDLE | | TEST |
| 1079 | FD24 | 56 | BTJO | %SETBIT, B, RSYNC8 | test for 1 |
| | FD25 | 08 | | | |
| | FD26 | 05 | | | |
| 1080 | FD27 | C5 | RSYNC7 | CLR B | TEST |
| 1081 | FD28 | 01 | IDLE | | TEST |
| 1082 | FD29 | 57 | BTJZ | %SETBIT, B, RSYNC9 | test for 0 |
| | FD2A | 08 | | | |
| | FD2B | 07 | | | |
| 1083 | FD2C | C5 | RSYNC8 | CLR B | TEST |
| 1084 | FD2D | 01 | IDLE | | TEST |
| 1085 | FD2E | 57 | BTJZ | %SETBIT, B, RSYNC7 | test for 0 |
| | FD2F | 08 | | | |
| | FD30 | F6 | | | |
| 1086 | FD31 | E0 | RESINK | JMP RESYNC | if two 1's in a row |
| | FD32 | 84 | | | |
| 1087 | * | | test for 2 more 0s | | |
| 1088 | FD33 | C5 | RSYNC9 | CLR B | TEST |
| 1089 | FD34 | 01 | IDLE | | TEST |
| 1090 | FD35 | 56 | BTJO | %SETBIT, B, RESINK | test for 1 |
| | FD36 | 08 | | | |
| | FD37 | F9 | | | |
| 1091 | FD38 | C5 | CLR | B | TEST |
| 1092 | FD39 | 01 | IDLE | | TEST |
| 1093 | FD3A | 56 | BTJO | %SETBIT, B, RESINK | test for 1 |
| | FD3B | 08 | | | |
| | FD3C | F4 | | | |
| 1094 | FD3D | 72 | MOV | %4, BITCON | bitcon will have been 1 |
| | FD3E | 04 | | | |
| | FD3F | 02 | | | |
| 1095 | FD40 | EE | TRAP | BIT8BA | wait end of nibble (spac |

1096 * register A will have been swapped (spacer only)
1097 * at this point the first byte for RCOMP is being picked up
1098 FD41 72 MOV %D03,COUNT length of file/record
FD42 03
FD43 2F
1099 FD44 72 MOV %NFILE,DATA# file & record numbers
FD45 3E
FD46 36
1100 FD47 A2 MOVP %2,DRIVE bit-sample margin test
FD48 02
FD49 06
1101 FD4A 0A RETS

1103 *-----
1104 * read wafer and compare to memory
1105 FD4B 73 RCOMPMM AND %RSTERR,FFLAG reset error flag
FD4C BF
FD4D 37
1106 FD4E EE RCMPPM2 TRAP BIT8BA wait end of byte
1107 * register A will have been swapped (data byte)
1108 FD4F 9D CMPA *DATAP compare data
FD50 36
1109 FD51 E2 JEQ RCMPPM3 test for match
FD52 03
1110 FD53 74 OR %FERROR,FFLAG set error flag
FD54 40
FD55 37
1111 FD56 D2 RCMPPM3 DEC DATAP dec data pointer
FD57 36
1112 FD58 E8 TRAP ADDCHK
1113 FD59 DA DJNZ COUNT,RCMPM2 add data to checksum
FD5A 2F dec byte counter
FD5B F2
1114 FD5C F0 TRAP BIT4BA wait end of byte
1115 * 1st nibble of RCOMPMM/RBYTE/RDL/RDLX/RCHKSM is being picked
1116 FD5D 42 MOV DLEN,NIBCON move DL to nibcon
FD5E 26
FD5F 2F
1117 FD60 42 MOV DLEN-1,NIBCON-1
FD61 25
FD62 2E
1118 FD63 0A RETS

1120 *-----
1121 * read DL & compare to memory
1122 FD64 78 RDL ADD %D04,BITCON pick up DL in bytes
FD65 04
FD66 02
1123 FD67 EE TRAP BIT8BA wait, bitcon=8, B->A
1124 * register A will have been swapped (data byte)
1125 FD68 D0 MOV A,NIBCON store ls byte of DL
FD69 2F
1126 FD6A EE TRAP BIT8BA wait, bitcon=8, B->A
1127 * register A will have been swapped (data byte)
1128 FD6B D0 MOV A,NIBCON-1 store ms byte of DL
FD6C 2E
1129 FD6D 7A SUB %D04,BITCON exit picking up nibble
FD6E 04
FD6F 02
1130 * this DL is compared to DL in SAB in main loop
1131 FD70 E0 JMP RDLX5
FD71 14

1133 *-----
1134 * read DL & transmit bus
1135 * HSK released in XREAD
1136 FD72 F0 RDLX TRAP BIT4BA wait, bitcon=4, B>A
1137 FD73 ED TRAP TBUS transmit data over bus
1138 FD74 D0 MOV A,NIBCON store 1s nibble
FD75 2F
1139 FD76 F0 TRAP BIT4BA wait, bitcon=4, B>A
1140 FD77 ED TRAP TBUS transmit data over bus
1141 FD78 B7 SWAP A position nibble
1142 FD79 44 OR A,NIBCON store 2nd nibble
FD7A 00
FD7B 2F
1143 FD7C F0 TRAP BIT4BA wait, bitcon=4, B>A
1144 FD7D ED TRAP TBUS transmit data over bus
1145 FD7E D0 MOV A,NIBCON-1 store 3rd nibble
FD7F 2E
1146 FD80 F0 TRAP BIT4BA wait, bitcon=4, B>A
1147 FD81 ED TRAP TBUS transmit data over bus
1148 FD82 B7 SWAP A position nibble
1149 FD83 44 OR A,NIBCON-1 store ms nibble
FD84 00
FD85 2E
1150 FD86 EE RDLX5 TRAP BIT8BA test for end of nibble
1151 * register A will have been swapped (spacer only)
1152 * non-data spacer followed by 1st nibble of data is picked up
1153 FD87 48 ADD NIBCON,CHKSUM add dl to checksum
FD88 2F
FD89 31
1154 FD8A 79 ADC %0,CHKSUM-1
FD8B 00
FD8C 30
1155 FD8D 48 ADD NIBCON-1,CHKSUM
FD8E 2E
FD8F 31
1156 FD90 79 ADC %0,CHKSUM-1
FD91 00
FD92 30
1157 FD93 DF RLC NIBCON multiply nibcon by 2
FD94 2F
1158 FD95 DF RLC NIBCON-1 (byte count -> nibble co
FD96 2E
1159 FD97 0A RETS
1160 *-----
1161 *RHELP TRAP REDRUM message terminated/slow
1162 * BR @EOM

```
1164      *-----  
1165      * read wafer & forget  
1166 FD98  F0  FORGT2 TRAP  BIT4BA      wait end of nibble  
1167 FD99  23      AND    %LSN,A      clear msn for checksum  
1168 FD9A  0F  
1169 FD9B  E8      TRAP    ADDCHK      add data to checksum  
116A FD9C  D2  FORGET  DEC    NIBCON      dec nibble counter  
116B FD9D  2F  
116C FD9E  78      SBB    .  %0,NIBCON-1  
116D FD9F  00  
116E FDA0  2E  
116F FDA1  E3      JC     FORGT2      test end of data  
1170 FDA2  F5  
1171 FDA3  E0      JMP    RBYTE3  
1172 FDA4  09
```

1174 *-----
1175 * read wafer and store
1176 FDA5 EE RBYTE TRAP BIT8BA wait end of byte
1177 * register A will have been swapped (data byte)
1178 FDA6 9B STA *DATAP store data
FDA7 36
1179 FDA8 D2 DEC DATAP dec data pointer
FDA9 36
1180 FDAA E8 TRAP ADDCHK add data to checksum
1181 FDAB DA DJNZ NIBCON, RBYTE dec byte counter
FDAC 2F
FDAD F7
1182 FDAE F0 RBYTE3 TRAP BIT4BA wait end of nibble
1183 * at this point the 1st nibble of RCHKSM is being picked up
1184 FDAF E0 JMP RCHKSM
FDB0 00

1186 *-----
1187 * read chekcsom
1188 FDB1 F4 RCHKSM TRAP BITEST wait end of nibble
1189 FDB2 72 MOV %4,BITCON restart bit count
FDB3 04
FDB4 02
1190 * 2nd half of first byte of checksum is being picked up
1191 FDB5 EE TRAP BIT8BA wait, bitcon=8, B->A
1192 * register A will have been swapped (data byte)
1193 FDB6 D0 MOV A,DLEN store ls byte checksum
FDB7 26
1194 FDB8 EE TRAP BIT8BA wait, bitcon=8, B->A
1195 * register A will have been swapped (data byte)
1196 FDB9 D0 MOV A,DLEN-1 store ms byte checksum
FDBA 25
1197 * turn off drive
1198 FDBB F5 TRAP TIMEX set up interrupt (time-o
1199 FDBC A2 MOVP %STOP, DRIVE motor off
FDBD FB
FDBE 06
1200 FDBF 12 MOV CCODE,A move command code to A
FDC0 20
1201 FDC1 D5 CLR STATUS assume ok status
FDC2 20
1202 FDC3 20 CMP %CREAD,A test for read code
FDC4 03
1203 FDC5 E2 JEQ RCHK52
FDC6 0B
1204 FDC7 20 CMP %CVERIFY,A test for verify code
FDC8 0C
1205 FDC9 E6 JNE RCHK53
FDCA 0E
1206 FDCB 77 BTJZ %FERROR,FFLAG,RCHK52 test for no verify error
FDCC 40
FDCC 37
FDCE 03
1207 FDCF 72 MOV %RSVERIFY,STATUS set up verify error
FDD0 18
FDD1 20
1208 FDD2 A2 RCHK52 MOVP %ZERO,BDATA begin return status/DL
FDD3 00
FDD4 10
1209 FDD5 A2 MOVP %DROP,BCNTL drop hsk
FDD6 01
FDD7 11
1210 FDD8 B4 INV A set command flag
1211 * verify checksum
1212 FDD9 40 RCHK53 CMP DLEN,CHKSUM compare checksum
FDDA 26
FDDB 31
1213 FDDC E6 JNE RCHK54 test checksum
FDDD 05
1214 FDDF 40 CMP DLEN-1,CHKSUM-1 compare checksum
FDE0 25
FDE0 30
1215 FDE1 E2 JEQ RLASTO test checksum
FDE2 06

| | | | | | | |
|------|-------|------|--------|-------|-----------------------|------------------------|
| 1216 | FDE3 | 27 | RCHK54 | BTJZ | %DFO, A, RCHKSE | test command flag |
| | FDE4 | FO | | | | |
| | FDE5 | 1B | | | | |
| 1217 | FDE6 | 72 | | MOV | %RSODATA, STATUS | set up data error |
| | FDE7 | 10 | | | | |
| | FDE8 | 2D | | | | |
| 1218 | | * | | | | |
| 1219 | | * | | | | |
| 1220 | FDE9 | 72 | RLAST0 | MOV | %DFA, NIBCON | 250 byte stop |
| | FDEA | FA | | | | |
| | FDEB | 2F | | | | |
| 1221 | FDEC | F1 | RLAST | TRAP | BIT10A | test for end |
| 1222 | FDED | A6 | | BTJOP | %EOTST, WAFER, RLAST2 | test for not EOT Found |
| | FDEE | 02 | | | | |
| | FDEF | 0A | | | | |
| | FDF0 | 03 | | | | |
| 1223 | FDF1 | 74 | | OR | %EOTFLG, FFLAG | set EOT flag |
| | FDF2 | 04 | | | | |
| | FDF3 | 37 | | | | |
| 1224 | FDF4 | DA | RLAST2 | DJNZ | NIBCON, RLAST | dec byte counter |
| | FDF5 | 2F | | | | |
| | FDF6 | F5 | | | | |
| 1225 | FDF7 | A2 | | MOV P | %STOP, DRIVE | turn drive off |
| | FDF8 | FB | | | | |
| | FDF9 | 06 | | | | |
| 1226 | F DFA | A2 | | MOV P | %HALT, TIMER | stop timer |
| | FDFB | 20 | | | | |
| | FDFC | 03 | | | | |
| 1227 | FDFD | A2 | | MOV P | %I123C, IOCNTL | clear INT1, 2&3 |
| | FDFE | AA | | | | |
| | FDFF | 00 | | | | |
| 1228 | FE00 | 0A | | RETS | | |
| 1229 | FE01 | 8E | RCHKSE | CALL | @RLAST0 | delay |
| | FE02 | FDE9 | | | | |
| 1230 | FE04 | FB | | TRAP | ERRWT | checksum error |
| 1231 | FE05 | 06 | | BYTE | RSDEVI | device error code |

1233 *-----
1234 * find EOT
1235 FE06 F5 FNDEOT TRAP TIMEX start timer operation
1236 FE07 72 MOV %D02,NIBCON-1 256 byte delay
1237 FE08 02
1238 FE09 2E
1239 FE0A 8E CALL @WSYNC5 call delay routine
1240 FE0B FBE8
1241 FE0D EF FEOT1 TRAP BITBAB wait end of nibble
1242 FE0E A6 BTJOP %EOTTST, WAFER, FEOT1 test for not EOT
1243 FE0F 02
1244 FE10 0A
1245 FE11 FB
1246 FE12 0A RETS EOT found
1247 *-----
1248 * start motor & timer operation
1249 FE13 06 ROLLEM DINT disable interrupts
1250 * COPY ALC, STRINGY, SRC, OLDWINT old microcode int.
1251 FE14 72 MOV %WRBIT1, INT2 set up INT2 opcode
1252 FE15 D6
1253 FE16 32
1254 FE17 72 MOV %WRBIT2, INT2+1 set up INT2 parameter
1255 FE18 02
1256 FE19 33
1257 FE1A 05 EINT re-enable ints
1258 FE1B A2 MOVP %BITIME, TIME half-bit time
1259 FE1C 0D
1260 FE1D 02
1261 FE1E 72 MOV %D08, BITCON start bit counter
1262 FE1F 08
1263 FE20 02
1264 FE21 A2 MOVP %I2CS, IOCNTL select & clear INT2
1265 FE22 8C
1266 FE23 00
1267 FE24 A2 MOVP %START, TIMER start timer
1268 FE25 A0
1269 FE26 03
1270 FE27 A2 MOVP %MT, DRIVE turn on motor
1271 FE28 F3
1272 FE29 06
1273 FE2A 0A RETS

1255 *-----
1256 * position wafer to file (assume valid file number present)
1257 FE2B 74 POSITN OR %FFOUND,FFLAG set found flag
FE2C 08
FE2D 37
1258 FE2E 77 BTJZ %LAST,FILE0,POSIT3 test for 0 not last file
FE2F 40
FE30 40
FE31 08
1259 FE32 76 BTJO %ACTIVE,FILE0,POSIT3 test for 0 active
FE33 80
FE34 40
FE35 07
1260 FE36 D3 INC NFILE no files present
FE37 3E
1261 FE38 73 POSIT1 AND %FMSSNG,FFLAG reset file-not-found
FE39 F7
FE3A 37
1262 FE3B E0 JMP FE0D3 file not found, find EOD
FE3C 2D
1263 FE3D 8E POSIT3 CALL @CMPFIL compare file header
FE3E FE84
1264 FE40 76 BTJO %FERROR,FFLAG,POSIT6 test for file not found
FE41 40
FE42 37
FE43 09
1265 FE44 77 BTJZ %FNAME,FFLAG,POSIT4 test file name/number
FE45 02
FE46 37
FE47 04
1266 FE48 77 BTJZ %ACTIVE,RECFIL-1,POSIT6 test for not active
FE49 80
FE4A 38
FE4B 01
1267 FE4C 0A POSIT4 RETS
1268 FE4D 8E POSIT6 CALL @FNDEOF find EOF
FE4E FE56
1269 FE50 76 BTJO %LAST,RECFIL-1,POSIT1 test for last file
FE51 40
FE52 38
FE53 E4
1270 FE54 E0 JMP POSIT3
FE55 E7

1272 *-----
1273 * find EOF
1274 FE56 EC FNDEOF TRAP EOFTST compare current/last rec
1275 FE57 E2 JZ FEOF2 test for not last record
1276 FE58 01
1276 FE59 0A RETS EOF found
1277 FE5A 8E FEOF2 CALL @RSINC inc rec#, read sync & hea
1278 FE5B FC9F
1278 FE5D EB TRAP RCOMPR compare file/record
1279 FE5E EA TRAP DEVERR test for device error
1280 FE5F 8E CALL @RDL read DL
1280 FE60 FD64
1281 * skip record
1282 FE62 8E CALL @FORGET read wafer & forget
1282 FE63 FD9C
1283 FE65 E0 JMP FNDEOF
1283 FE66 EF

1285 *-----
1286 * find EOF of last file (assume valid file number present)
1287 FE67 74 FNDEOD OR %FFOUND,FFLAG set found flag
FE68 08
FE69 37
1288 FE6A 32 FE0D3 MOV NFILE,B set up index
FE6B 3E
1289 FE6C F6 TRAP FILDAT load file parameters
1290 FE6D 77 BTJZ %LAST,RECFIL-1,FE0D2 test for not last file
FE6E 40
FE6F 38
FE70 08
1291 FE71 76 BTJO %ACTIVE,RECFIL-1,FE0D2 test for active
FE72 80
FE73 38
FE74 07
1292 * if last file is inactive it must be file 0
1293 FE75 73 AND %FMSSNG,FFLAG reset file not found
FE76 F7
FE77 37
1294 FE78 0A RETS
1295 FE79 8E FE0D1 CALL @CMPPFIL read next file header
FE7A FE84
1296 FE7C 8E FE0D2 CALL @FNDEOF find EOF
FE7D FE56
1297 FE7F 77 BTJZ %LAST,RECFIL-1,FE0D1 test for not last file
FE80 40
FE81 38
FE82 F6
1298 FE83 0A RETS

| | | | | | | |
|-----------|--|--------|------|-----------------------|------------------------|--|
| 1300 | *- | | | | | |
| 1301 | * compare file header (tests file/record & filename) | | | | | |
| 1302 FE84 | D3 | CMPFIL | INC | NFILE | next file | |
| FE85 | 3E | | | | | |
| 1303 FE86 | 32 | MOV | | NFILE, B | load file parameters | |
| FE87 | 3E | | | | | |
| 1304 FE88 | F6 | TRAP | | FILDAT | load file parameters | |
| 1305 FE89 | 72 | MOV | | %DOC, DLEN | filename DL | |
| FE8A | 0C | | | | | |
| FE8B | 26 | | | | | |
| 1306 FE8C | D5 | CLR | | NREC | record 0 | |
| FE8D | 3D | | | | | |
| 1307 FE8E | D5 | CLR | | NREC-1 | | |
| FE8F | 3C | | | | | |
| 1308 FE90 | 8E | CALL | | @RSYNC | read sync | |
| FE91 FCA4 | | | | | | |
| 1309 FE93 | EB | TRAP | | RCOMPRESS | compare file/record | |
| 1310 FE94 | EA | TRAP | | DEVERR | test for device error | |
| 1311 FE95 | 72 | MOV | | %FNAME1, DATAP | set up data pointer | |
| FE96 | 21 | | | | | |
| FE97 | 36 | | | | | |
| 1312 FE98 | F4 | TRAP | | BITEST | restart bit count | |
| 1313 FE99 | 72 | MOV | | %4, BITCON | | |
| FE9A | 04 | | | | | |
| FE9B | 02 | | | | | |
| 1314 | * 2nd half of 1st filename byte is being picked up | | | | | |
| 1315 FE9C | 76 | BTJO | | %FNAME, FFLAG, CMPFL4 | test file name/number | |
| FE9D | 02 | | | | | |
| FE9E | 37 | | | | | |
| FE9F | 0E | | | | | |
| 1316 FEA0 | 73 | AND | | %RSTERR, FFLAG | clear error flag | |
| FEA1 | BF | | | | | |
| FEA2 | 37 | | | | | |
| 1317 FEA3 | 4D | CMP | | RNUM, NFILE | compare current/target | |
| FEA4 | 2A | | | | | |
| FEA5 | 3E | | | | | |
| 1318 FEA6 | E2 | JEQ | | CMPFL3 | test for match | |
| FEA7 | 03 | | | | | |
| 1319 FEA8 | 74 | OR | | %FERROR, FFLAG | set error flag | |
| FEA9 | 40 | | | | | |
| FEAA | 37 | | | | | |
| 1320 FEA8 | 8C | CMPFL3 | BR | @RBYTE | read file name | |
| FEAC FDB5 | | | | | | |
| 1321 FEA8 | EB | CMPFL4 | TRAP | RCOMPRESS | compare filename | |
| 1322 FEA8 | 8C | | BR | @RCHKSM | read checksum | |
| FEB0 FDB1 | | | | | | |

| | | | | |
|-----------|----------------|--------|------|--------------------------|
| 1324 | ----- | | | |
| 1325 | * test for EOF | | | |
| 1326 FEB2 | 73 | TSTEOF | AND | %RSTEOF, FFLAG |
| FEB3 | 7F | | | reset EOF flag |
| FEB4 | 37 | | | |
| 1327 FEB5 | 4D | | CMP | NREC, RECFIL |
| FEB6 | 3D | | | compare current/last rec |
| FEB7 | 39 | | | |
| 1328 FEB8 | E6 | | JNE | TEOF2 |
| FEB9 | 0C | | | |
| 1329 FEBA | 32 | | MOV | RECFIL-1, B |
| FEBB | 38 | | | |
| 1330 FEBC | 53 | | AND | %EOF, B |
| FEBD | 0F | | | |
| 1331 FEBE | 3D | | CMP | NREC-1, B |
| FEBF | 3C | | | |
| 1332 FEC0 | E6 | | JNE | TEOF2 |
| FEC1 | 04 | | | |
| 1333 FEC2 | 74 | | OR | %EOFFLG, FFLAG |
| FEC3 | 80 | | | set EOF flag |
| FEC4 | 37 | | | |
| 1334 FEC5 | 0A | | RETS | |
| 1335 FEC6 | C5 | TEOF2 | CLR | B |
| 1336 FEC7 | 0A | | RETS | status (JNZ/JZ: EOF/not) |

| | | | | |
|------|------|------------------|----------------------------------|--------------------------|
| 1338 | | * | | |
| 1339 | | * more in CTABLX | | |
| 1340 | FEC8 | 76 | TSTB10 BTJO %>FF, BITCON, TSTB10 | wait end of byte |
| | FEC9 | FF | | |
| | FECA | 02 | | |
| | FECB | FC | | |
| 1341 | FECC | 72 | MOV %>10, BITCON | restart bit-count |
| | FECD | 10 | | |
| | FECE | 02 | | |
| 1342 | FECF | C0 | MOV A, B | move data through queue |
| 1343 | FED0 | 0A | RETS | |
| 1344 | FED1 | 76 | TSTB8A BTJO %>FF, BITCON, TSTB8A | wait for end of nibble |
| | FED2 | FF | | |
| | FED3 | 02 | | |
| | FED4 | FC | | |
| 1345 | FED5 | 72 | MOV %>8, BITCON | restart nibble bit-count |
| | FED6 | 08 | | |
| | FED7 | 02 | | |
| 1346 | FED8 | C0 | MOV A, B | move data through queue |
| 1347 | FED9 | 0A | RETS | |
| 1348 | FEDA | 76 | TSTB8B BTJO %>FF, BITCON, TSTB8B | wait end of byte |
| | FEDB | FF | | |
| | FEDC | 02 | | |
| | FEDD | FC | | |
| 1349 | FEDE | 72 | MOV %>08, BITCON | restart bit counter |
| | FEDF | 08 | | |
| | FEE0 | 02 | | |
| 1350 | FEE1 | 62 | MOV B, A | move data through queue |
| 1351 | FEE2 | C5 | CLR B | clear input bits |
| 1352 | FEE3 | B7 | SWAP A | position nibbles |
| 1353 | FEE4 | 0A | RETS | |

1355 *-----
1356 * store file parameters
1357 FEE5 CE STFIELD RL B adjust index
1358 FEE6 CE RL B
1359 FEE7 12 STFIL2 MOV MAXLEN,A
FEE8 38
1360 FEE9 AB STA @FILE0+3(B) store ls byte of MRL
FEEA 0043
1361 FEEC 12 MOV MAXLEN-1,A
FEED 3A
1362 FEEE AB STA @FILE0+2(B) store ms byte of MRL
FEFF 0042
1363 FEF1 12 MOV RECFIL,A
FEF2 39
1364 FEF3 AB STA @FILE0+1(B) store ls byte of # record
FEF4 0041
1365 FEF6 12 MOV RECFIL-1,A
FEF7 38
1366 FEF8 AB STA @FILE0(B) store ms byte of # record
FEF9 0040
1367 FEFB 0A RETS
1368 *-----
1369 * load file parameters
1370 FEFC CE LDFILD RL B adjust index
1371 FEFD CE RL B
1372 FEFE AA LDA @FILE0+3(B) load ls byte of MRL
FEFF 0043
1373 FF01 D0 MOV A, MAXLEN
FF02 38
1374 FF03 AA LDA @FILE0+2(B) load ms byte of MRL
FF04 0042
1375 FF06 D0 MOV A, MAXLEN-1
FF07 3A
1376 FF08 AA LDA @FILE0+1(B) load ls byte of # record
FF09 0041
1377 FF0B D0 MOV A, RECFIL
FF0C 39
1378 FF0D AA LDA @FILE0(B) load ms byte of # record
FF0E 0040
1379 FF10 D0 MOV A, RECFIL-1
FF11 38
1380 FF12 0A RETS

1382 *-----
1383 * miscellaneous traps
1384 * let HSK float
1385 FF13 A2 FLOAT MOVP %HSKSET, BCNTL frequent operation
FF14 00
FF15 11
1386 FF16 0A RETS
1387 * add A to checksum
1388 FF17 48 ADACHK ADD A, CHKSUM operant frequation
FF18 00
FF19 31
1389 FF1A 79 ADC %0, CHKSUM-1
FF1B 00
FF1C 30
1390 FF1D 0A RETS
1391 * device error handler
1392 FF1E 76 LOST BTJO %FERROR, FFLAG, LOST2 test for device error
FF1F 40
FF20 37
FF21 01
1393 FF22 0A RETS no device error
1394 FF23 FA LOST2 TRAP REDRUM
1395 FF24 FB TRAP ERRWT error
1396 FF25 06 BYTE RSDEVI device error

1398 *-----
1399 * receive and store data from bus
1400 FF26 E9 RNXHSK TRAP TFLOAT let HSK float
1401 FF27 BE RCVPAB CALL @RCVNIB receive lsd
FF28 FF4F
1402 FF2A E9 TRAP TFLOAT let HSK float
1403 FF2B 62 MOV B,A save lsd
1404 FF2C BE CALL @RCVNIB receive msd, hold HSK
FF2D FF4F
1405 FF2F C7 SWAP B justify byte
1406 FF30 64 OR B,A combine nibbles
1407 FF31 D5 CLR DATAP-1 keep pointer in reg file
FF32 35
1408 * this is the only CLR DATAP-1
1409 FF33 98 STA *DATAP store byte
FF34 36
1410 FF35 D2 DEC DATAP decrement pointer
FF36 36
1411 FF37 D2 DEC COUNT decrement counter
FF38 2F
1412 FF39 E3 JC RNXHSK test for end of data
FF3A EB
1413 FF3B 0A RETS
1414 *-----
1415 * load and transmit data to bus
1416 FF3C E9 WNXHSK TRAP TFLOAT let HSK float
1417 FF3D 9A LDA *DATAP load data into a
FF3E 36
1418 FF3F C0 MOV A,B move nibble into b
1419 FF40 BE CALL @XMTNIB transmit lsd
FF41 FF62
1420 FF43 E9 WNXPA2 TRAP TFLOAT let HSK float
1421 FF44 C7 SWAP B position data
1422 FF45 8E CALL @XMTNIB transmit msd, hold HSK
FF46 FF62
1423 FF48 D2 DEC DATAP decrement pointer
FF49 36
1424 FF4A D2 DEC COUNT decrement counter
FF4B 2F
1425 FF4C E3 JC WNXHSK test for end of data
FF4D EE
1426 FF4E 0A RETS

1428 *-----
1429 * receive nibble from bus
1430 FF4F A7 RCVNIB BTJZP %BAV, BSTAT, GONE test for BAV active
FF50 02
FF51 11
FF52 0C
1431 FF53 A7 BTJZP %IRQ, BSTAT, RCVNIB test for bus data ready
FF54 08
FF55 11
FF56 F8
1432 FF57 A2 MOVP %RELEAS, BCNTL reset IRQ
FF58 01
FF59 11
1433 FF5A 91 MOVP BDATA, B read data from bus
FF5B 10
1434 FF5C 53 AND %LSN, B clear msn
FF5D 0F
1435 FF5E OA RETS
1436 *-----
1437 * disappearing-message-frame handler
1438 FF5F 8C GONE BR @EOM message terminated
FF60 F808
1439 *-----
1440 * transmit nibble to bus
1441 FF62 A7 XMTNIB BTJZP %BAV, BSTAT, GONE test for BAV active
FF63 02
FF64 11
FF65 F9
1442 FF66 A6 BTJOP %HSK, BSTAT, XMTNIB test for bus ready
FF67 01
FF68 11
FF69 F8
1443 FF6A 92 MOVP B, BDATA send data over bus
FF6B 10
1444 FF6C A2 MOVP %DROP, BCNTL drop HSK
FF6D 01
FF6E 11
1445 FF6F OA RETS

1447 *-----
1448 * transmit nibble over bus
1449 FF70 A7 XMTBUS BTJZP %BAV,BSTAT,GONE test for BAV active
FF71 02
FF72 11
FF73 E8
1450 FF74 A6 BTJOP %HSK,BSTAT,GONE test for bus ready
FF75 01
FF76 11
FF77 E7
1451 FF78 82 MOVP A,BDATA send data over bus
FF79 10
1452 FF7A A2 MOVP %DROP,BCNTL drop HSK
FF7B 01
FF7C 11
1453 FF7D E9 TRAP TFLOAT let HSK float
1454 FF7E 0A RETS
1455 *-----
1456 * interrupt 1 routine for wake-up
1457 FF7F A2 WAKEUP MOVP %I1230,IOCNTL clear INT1,2&3
FF80 AA
FF81 00
1458 * joins Isync2 below
1459 FF82 A2 MOVP %HALT,TIMER
FF83 20
FF84 03
1460 *-----
1461 * interrupt 2 routine for restart bit time sync
1462 FF85 0B ISYNC2 RETI
1463 * COPY ALC.STRINGY.SRC.OLDINT old microcode int.
1464 *-----
1465 * interrupt 3 routine for bit time sync
1466 FF86 A2 ISYNC3 MOVP %START,TIMER restart timer for sync
FF87 A0
FF88 03
1467 FF89 A2 MOVP %I3C23S,IOCNTL clear INT3,select 2&3
FF8A B4
FF8B 00
1468 FF8C 0B RETI
1469 *-----
1470 FF8D 76 TSTBIT BTJO %>FF,BITCON,TSTBIT wait end of nibble/byte
FF8E FF
FF8F 02
FF90 FC
1471 FF91 0A RETS

| | | | | |
|----------------|------------------------------------|------|-------------------|--------------------------|
| 1473 | * | | | |
| 1474 | * first-pass command-decode table | | | |
| 1475 FF92 F9BC | CTABLE | DATA | XCL0DD | 00 open: no open;read BL |
| 1476 FF94 FA58 | | DATA | XOPNDR | 01 close: open |
| 1477 FF96 FA58 | | DATA | XOPNDR *UNSUPP | 02 close/delete: open |
| 1478 FF98 FA58 | | DATA | XOPNDR | 03 read: open |
| 1479 FF9A FA58 | | DATA | XOPNDR | 04 write: open |
| 1480 FF9C FA58 | | DATA | XOPNDR | 05 position record: open |
| 1481 FF9E F9CA | | DATA | XCL0DD *UNSUPP | 06 delete: no open;read |
| 1482 FFA0 F9AB | | DATA | XRSTAT | 07 return status |
| 1483 FFA2 F86C | | DATA | UNSUPP | 08 unsupported svreq en |
| 1484 FFA4 F86C | | DATA | UNSUPP | 09 unsupported svreq di |
| 1485 FFA6 FFCE | | DATA | SNOTME | 0A was it you |
| 1486 FFA8 F86C | | DATA | UNSUPP | 0B unsupported you are m |
| 1487 FFAA FA58 | | DATA | XOPNDR | 0C verify: open |
| 1488 FFAC F9F2 | | DATA | XCL0DF | 0D format: no open;skip |
| 1489 FFAE F9EF | | DATA | XCL0DC | 0E catalog: no open |
| 1490 | * | | | |
| 1491 | * second-pass command-decode table | | | |
| 1492 FFB0 FA60 | CTABLX | DATA | XOPEN | 00 open address |
| 1493 FFB2 F89E | | DATA | XCLOSE | 01 close address |
| 1494 FFB4 F8B5 | | DATA | XCLSDL | 02 delete open file |
| 1495 FFB6 F97B | | DATA | XREAD | 03 read data address |
| 1496 FFB8 F8F5 | | DATA | XWRITE | 04 write data address |
| 1497 FFB8 F8DF | | DATA | XPOSIT | 05 position record addre |
| 1498 FFBC F8BE | | DATA | XDELET *FREE | 06 delete address |
| 1499 | * | | | |
| 1500 | * unused space in table | | | |
| 1501 FFBE 76 | TSTB4B | BTJO | %FF,BITCON,TSTB4B | wait end of nibble |
| FFBF FF | | | | |
| FFC0 02 | | | | |
| FFC1 FC | | | | |
| 1502 FFC2 72 | MOV | | %4,BITCON | restart nibble bit-count |
| FFC3 04 | | | | |
| FFC4 02 | | | | |
| 1503 FFC5 62 | MOV | | B,A | move data through queue |
| 1504 FFC6 C5 | CLR | | B | set all input bits to 0 |
| 1505 FFC7 0A | RETS | | | |
| 1506 | * | | | |
| 1507 FFC8 F941 | DATA | | XVERIF | 0C verify address |
| 1508 FFCA FB37 | DATA | | XFORMA | 0D format address |
| 1509 FFCC FB48 | DATA | | XCATAL | 0E catalog address |

| | | | | |
|----------------|-----------|---------|--------|--|
| 1511 | *----- | | | |
| 1512 FFCE FB | SNOTME | TRAP | ERRWT | error |
| 1513 FFCF OA | BYTE | | RSNOTI | itwasntme |
| 1514 | *----- | | | |
| 1515 | * vectors | | | |
| 1516 FFDO | AORG | >FFDO | | |
| 1517 FFDO FF17 | DATA | ADACHK | | trap 23 add A to checksu |
| 1518 FFD2 FF13 | DATA | FLOAT | | trap 22 let HSK float |
| 1519 FFD4 FF1E | DATA | LOST | | trap 21 device error han |
| 1520 FFD6 FD4B | DATA | RCOMPM | | trap 20 compare wafer/RA |
| 1521 FFD8 FEB2 | DATA | TSTE0F | | trap 19 test for fOF |
| 1522 FFDA FF70 | DATA | XMTBUS | | trap 18 transmit data ov |
| 1523 FFDC FEDA | DATA | TSTB8B | | trap 17 wait, bitcon=8 B |
| 1524 FFDE FED1 | DATA | TSTB8A | | trap 16 wait, bitcon=8 A |
| 1525 FFE0 FFBF | DATA | TSTB4B | | trap 15 wait, bitcon=4 B |
| 1526 FFE2 FEC8 | DATA | TSTB10 | | trap 14 wait, bitcon=10 |
| 1527 FFE4 FEES | DATA | STFIELD | | trap 13 store file data |
| 1528 FFE6 FC73 | DATA | TSTEOT | | trap 12 test for EOT |
| 1529 FFE8 FF8D | DATA | TSTBIT | | trap 11 test end of bit |
| 1530 FFEA FE13 | DATA | ROLLEM | | trap 10 start wafer oper |
| 1531 FFEC FEFC | DATA | LDFILD | | trap 09 get file data |
| 1532 FFEF FF26 | DATA | RNXHSK | | trap 08 receive PAB/data |
| 1533 FFF0 FF86 | DATA | ISYNC3 | | *not in IC trap 07 ampl defaults to |
| 1534 FFF2 FF3C | DATA | WNXHSK | | trap 06 transmit respons |
| 1535 FFF4 FBFA | DATA | KILL | | trap 05 kill open file |
| 1536 FFF6 F86E | DATA | WTERR | | trap 04 error handler |
| 1537 FFF8 FF86 | DATA | ISYNC3 | | interrupt 3 |
| 1538 FFFA 0032 | DATA | INT2 | | interrupt 2 |
| 1539 FFFC FF7F | DATA | WAKEUP | | interrupt 1 |
| 1540 FFFE F806 | DATA | INIT | | reset |

NO ERRORS, NO WARNINGS

| MT358F10 | MLP | FAMILY | ASSEMBLER | 1.0 | | 12:43:22 | 8/ 9/82 | |
|----------|-------|--------|------------|------|------|----------|---------|-----------|
| LABEL | VALUE | DEFN | REFERENCES | | | | | PAGE 0069 |
| ACTIVE | 0080 | 0241 | 1259 | 1266 | 1291 | | | |
| ADACHK | FF17 | 1388 | 1517 | | | | | |
| ADDCHK | 0017 | 0340 | 0578 | 0864 | 0951 | 1052 | 1112 | 1168 |
| AREG | 0000 | 0054 | | | | | | |
| ATTRIB | 0022 | 0064 | 0449 | 0493 | 0562 | 0656 | 0657 | 0660 |
| | | | 0757 | 0763 | | | | |
| BATTRY | 0040 | 0216 | 0883 | 1023 | | | | |
| BAV | 0002 | 0232 | 0920 | 1430 | 1441 | 1449 | | |
| BCNTL | 0111 | A0012 | 0371 | 0441 | 0922 | 0970 | 1209 | 1385 |
| BDATA | 0110 | A0011 | 0923 | 0969 | 1208 | 1433 | 1443 | 1451 |
| BIT10A | 000E | 0342 | 0860 | 0867 | 0929 | 0936 | 0938 | 0959 |
| BIT4BA | 000F | 0343 | 0546 | 0575 | 1114 | 1136 | 1139 | 1143 |
| BITBAH | 0010 | 0344 | 0880 | 0894 | 0947 | 1238 | | |
| BITBAA | 0011 | 0345 | 1095 | 1106 | 1123 | 1126 | 1150 | 1176 |
| BITCON | 0002 | 0056 | 0650 | 0939 | 0965 | 1019 | 1032 | 1053 |
| | | | 1122 | 1129 | 1189 | 1249 | 1313 | 1340 |
| | | | 1348 | 1349 | 1470 | 1501 | 1502 | |
| BITEST | 000B | 0341 | 0649 | 0974 | 1021 | 1188 | 1312 | |
| BITIME | 000D | B0002 | | | | | | |
| BLANK | 0020 | 0278 | 0619 | | | | | |
| BLEN | 0028 | 0067 | 0799 | 0800 | | | | |
| BLOPEN | 0024 | 0065 | 0595 | 0607 | 0730 | 0735 | 0737 | 0746 |
| | | | 0771 | 0773 | 0776 | 0779 | 0780 | 0768 |
| BOSTIM | 0067 | B0011 | 1017 | | | | | |
| BOTIME | 00B6 | B0012 | 0644 | | | | | |
| BREG | 0001 | 0055 | | | | | | |
| BROPC | 008C | 0279 | 0434 | 1028 | | | | |
| BSTAT | 0111 | A0013 | 0440 | 0920 | 0921 | 1430 | 1431 | 1441 |
| CAPTUR | 0003 | A0004 | 1040 | | | | | |
| CCATAL | 000E | 0288 | | | | | | |
| CCCLOSE | 0001 | 0290 | | | | | | |
| CCLSDL | 0002 | 0291 | | | | | | |
| CCODE | 002C | 0070 | 0075 | 0388 | 0455 | 0633 | 0653 | 0666 |
| | | | 1200 | | | | | |
| CDELET | 0006 | 0292 | | | | | | |
| CFORMA | 000D | 0289 | | | | | | |
| CHKBUS | FC0E | 0918 | 0550 | | | | | |
| CHKSUM | 0031 | 0078 | 0079 | 0080 | 0891 | 0892 | 0918 | 0919 |
| | | | 0934 | 0961 | 0963 | 1034 | 1035 | 1058 |
| | | | 1063 | 1065 | 1068 | 1069 | 1153 | 1154 |
| | | | 1214 | 1388 | 1389 | | 1155 | 1156 |
| CMPFIL | FE84 | 1302 | 1263 | 1295 | | | | |
| CMPFL3 | FEAB | 1320 | 1318 | | | | | |
| CMPFL4 | FEAE | 1321 | 1315 | | | | | |
| CNULL | 00FE | 0293 | 0389 | | | | | |
| COPEN | 0000 | 0294 | | | | | | |
| COUNT | 002F | 0077 | 0377 | 0386 | 0407 | 0412 | 0519 | 0584 |
| | | | 0742 | 0745 | 0751 | 0807 | 0811 | 0814 |
| | | | 0866 | 1098 | 1113 | 1411 | 1424 | 0819 |
| CPOSIT | 0005 | 0297 | | | | | | |
| CREAD | 0003 | 0295 | 1202 | | | | | |
| CRESET | 00FF | 0296 | 0394 | | | | | |
| CRSTAT | 0007 | 0298 | | | | | | |
| CTABLE | FF92 | 1475 | 0399 | 0401 | | | | |
| CTABLX | FFB0 | 1492 | 0671 | 0673 | | | | |
| CVERIF | 000C | 0299 | 1204 | | | | | |
| CWRITE | 0004 | 0300 | 0967 | 0989 | | | | |

| MT358F10 | MLP | FAMILY | ASSEMBLER | 1.0 | | 12:43:22 | 8/ 9/82 | |
|----------|-------|--------|--|--|--|--|---|-----------|
| LABEL | VALUE | DEFN | REFERENCES | | | | | PAGE 0070 |
| DATAP | 0036 | 0081 | 0082 0378 0410 0413 0520 0607 0622 0648 0707 | 0743 0746 0752 0806 0810 0813 0818 0824 0862 | 0863 0890 1005 1099 1108 1111 1178 1177 1311 | 1407 1409 1410 1417 1423 | | |
| DATFIL | 000D | 0346 | 0452 0467 0906 | | | | | |
| DCODE | 002D | 0071 | 0072 | | | | | |
| DDRD | 000B | A0010 | 0849 1020 | | | | | |
| DEVERR | 0015 | 0347 | 0535 0570 0647 1279 1310 | | | | | |
| DFORMA | 0040 | 0242 | 0791 | | | | | |
| DIRECT | 003F | 0087 | | | | | | |
| DISPLAY | 0010 | 0243 | 0760 | | | | | |
| DLEN | 0026 | 0066 | 0408 0410 0496 0508 0511 0513 0514 0517 | 0520 0531 0532 0539 0541 0597 0603 0611 0612 | 0643 0703 0740 0741 0743 0804 0805 0806 0869 | 0870 0931 0933 0935 0937 1003 1116 1117 1193 | 1196 1212 1214 1305 | |
| DRIVE | 0006 | A0008 | 0372 0376 0424 0833 0846 0848 0976 0978 1100 | 1199 1225 1252 | | | | |
| DROP | 0001 | 0225 | 0970 1209 1444 1452 | | | | | |
| ENDRCT | 007F | 0104 | 0648 1005 | | | | | |
| E0FFLG | 0080 | 0244 | 1333 | | | | | |
| E0FTST | 0013 | 0348 | 0526 0565 0591 0739 1274 | | | | | |
| E0M | F808 | 0367 | 0385 0390 0415 0587 1438 | | | | | |
| E0M2 | F86A | 0415 | 0446 0455 | | | | | |
| E0TCHK | 000C | 0349 | 0865 0881 0949 | | | | | |
| E0TFLG | 0004 | 0245 | 0504 0981 1223 | | | | | |
| E0TTST | 0002 | 0217 | 0985 1036 1222 1239 | | | | | |
| ERRWT | 0004 | 0350 | 0419 0462 0494 0497 0528 0543 0563 0605 0631 | 0658 0662 0680 0690 0696 0733 0761 0766 0802 | 0877 0913 0991 1037 1230 1395 1512 | | | |
| FE0D1 | FE79 | 1295 | 1297 | | | | | |
| FE0D2 | FE7C | 1296 | 1290 1291 | | | | | |
| FE0D3 | FE6A | 1288 | 1262 | | | | | |
| FE0F2 | FE5A | 1277 | 1275 | | | | | |
| FE0T1 | FE0D | 1238 | 1239 | | | | | |
| FERROR | 0040 | 0246 | 0553 1110 1206 1264 1319 1392 | | | | | |
| FFLAG | 0037 | 0082 | 0083 0366 0391 0393 0446 0448 0450 0460 0461 | 0484 0486 0492 0504 0553 0592 0624 0628 0630 | 0661 0668 0679 0686 0700 0750 0847 0902 0903 | 0981 1105 1110 1206 1223 1257 1261 1264 1265 | 1287 1293 1315 1316 1319 1326 1333 1392 | |
| FFOUND | 0008 | 0247 | 0486 0668 0686 0700 1257 1287 | | | | | |
| FILDAT | 0009 | 0351 | 0711 0720 1289 1304 | | | | | |
| FILE0 | 0040 | 0088 | 0474 0477 0791 1258 1259 1360 1362 1364 1366 | 1372 1374 1376 1378 | | | | |
| FILE1 | 0044 | 0089 | | | | | | |
| FILE2 | 0048 | 0090 | | | | | | |
| FILE3 | 004C | 0091 | | | | | | |
| FILE4 | 0050 | 0092 | | | | | | |
| FILE5 | 0054 | 0093 | | | | | | |
| FILE6 | 0058 | 0094 | | | | | | |
| FILE7 | 005C | 0095 | | | | | | |
| FILE8 | 0060 | 0096 | | | | | | |
| FILE9 | 0064 | 0097 | | | | | | |
| FILEA | 0068 | 0098 | | | | | | |
| FILEB | 006C | 0099 | | | | | | |
| FILEC | 0070 | 0100 | | | | | | |

MT358F10 MLP FAMILY ASSEMBLER 1.0
LABEL VALUE DEFN REFERENCES

12:43:22 8/9/82

| MT358F10 LABEL | MLP VALUE | FAMILY DEFN | ASSEMBLER REFERENCES | 1.0 | | | | | | 12:43:22 | 8/ 9/82 | PAGE 0072 |
|-------------------|--------------|----------------|-------------------------|------|------|------|------|------|------|----------|---------|-----------|
| LAST | 0040 | 0254 | 0468 1297 | 0476 | 0693 | 0695 | 0712 | 0765 | 1258 | 1269 | 1290 | |
| LASTO | 00BF | 0263 | 0469 | 0713 | 0721 | | | | | | | |
| LDFILD | FEFC | 1370 | 1531 | | | | | | | | | |
| LOST | FF1E | 1392 | 1519 | | | | | | | | | |
| LOST2 | FF23 | 1394 | 0487 | 1392 | | | | | | | | |
| LRANGE | 0038 | B0005 | 1044 | | | | | | | | | |
| LSN | 000F | 0282 | 0547 | 0577 | 0817 | 0925 | 1167 | 1434 | | | | |
| LUNO | 002B | 0069 | 0074 | | | | | | | | | |
| MAX | 00BF | B0006 | 0436 | 1018 | | | | | | | | |
| MAXLEN | 003B | 0084 | 0085 | 0508 | 0511 | 0513 | 0514 | 0726 | 0727 | 0770 | 0771 | |
| | | | 0773 | 0776 | 0779 | 0780 | 0818 | 1359 | 1361 | 1373 | 1375 | |
| MOOSE | F845 | 0394 | 0392 | | | | | | | | | |
| MT | 00F3 | 0208 | 0846 | 1252 | | | | | | | | |
| MTBAR | 0008 | 0209 | 0977 | | | | | | | | | |
| MTWE | 00F1 | 0210 | 0848 | | | | | | | | | |
| NEWFIL | 00C0 | 0255 | 0724 | | | | | | | | | |
| NEWFLG | 0014 | 0256 | 0750 | | | | | | | | | |
| NFILE | 003E | 0086 | 0451 | 0465 | 0482 | 0640 | 0698 | 0701 | 0717 | 0810 | 0839 | |
| | | | 0882 | 0890 | 0905 | 1000 | 1099 | 1260 | 1288 | 1302 | 1303 | |
| | | | | 1317 | | | | | | | | |
| NIBCON | 002F | 0076 | 0077 | 0078 | 0539 | 0541 | 0554 | 0555 | 0579 | 0580 | 0835 | |
| | | | 0837 | 0843 | 0851 | 0869 | 0870 | 0884 | 0885 | 0893 | 0895 | |
| | | | 0896 | 0944 | 0945 | 0952 | 0953 | 1116 | 1117 | 1120 | 1128 | |
| | | | 1138 | 1142 | 1145 | 1149 | 1153 | 1155 | 1157 | 1158 | 1169 | |
| | | | 1170 | 1181 | 1220 | 1224 | 1236 | | | | | |
| NMLEN | 000C | 0060 | 0061 | 0062 | 0814 | | | | | | | |
| NREC | 003D | 0085 | 0086 | 0499 | 0500 | 0505 | 0507 | 0641 | 0642 | 0704 | 0705 | |
| | | | 0752 | 0840 | 0841 | 1001 | 1002 | 1014 | 1015 | 1304 | 1307 | |
| | | | | 1327 | 1331 | | | | | | | |
| OPNRST | 00EF | 0264 | 0448 | 0460 | 0903 | | | | | | | |
| ORANGE | 0006 | B0007 | 1051 | | | | | | | | | |
| POSIT1 | FE38 | 1261 | 1269 | | | | | | | | | |
| POSIT3 | FE3D | 1263 | 1258 | 1259 | 1270 | | | | | | | |
| POSIT4 | FE4C | 1267 | 1265 | | | | | | | | | |
| POSIT6 | FE4D | 1268 | 1264 | 1266 | | | | | | | | |
| POSITN | FE2B | 1257 | 0485 | 0665 | | | | | | | | |
| PSCALE | 0003 | A0005 | | | | | | | | | | |
| RBUS | FC14 | 0920 | 0950 | | | | | | | | | |
| RBYTE | FDA5 | 1176 | 0652 | 1181 | 1320 | | | | | | | |
| RBYTE3 | FDAE | 1182 | 0557 | 0582 | 1172 | | | | | | | |
| RBYTEJ | F976 | 0557 | | | | | | | | | | |
| RCHKS2 | FDD2 | 1208 | 1203 | 1206 | | | | | | | | |
| RCHKS3 | FDD9 | 1212 | 1205 | | | | | | | | | |
| RCHKS4 | FDE3 | 1216 | 1213 | | | | | | | | | |
| RCHKSE | FE01 | 1229 | 1216 | | | | | | | | | |
| RCHKSM | FDB1 | 1188 | 1184 | 1322 | | | | | | | | |
| RCMPB2 | F960 | 0546 | 0556 | | | | | | | | | |
| RCMPM2 | FD4E | 1106 | 1113 | | | | | | | | | |
| RCMPM3 | FD56 | 1111 | 1109 | | | | | | | | | |
| RCOMPB | F96F | 0554 | 0542 | 0552 | | | | | | | | |
| RCOMPMP | FD4B | 1105 | 1520 | | | | | | | | | |
| RCOMPB | 0014 | 0352 | 0534 | 0569 | 0646 | 1278 | 1309 | 1321 | | | | |
| RCVCNT | 0008 | 0353 | 0387 | 0609 | 0623 | | | | | | | |
| RCVDMO | F878 | 0430 | 0442 | | | | | | | | | |
| RCVDM1 | F88D | 0439 | 0440 | | | | | | | | | |
| RCVNIB | FF4F | 1430 | 1401 | 1404 | 1431 | | | | | | | |

MT358F10 MLP FAMILY ASSEMBLER 1.0
LABEL VALUE DEEN REFERENCES

12:43:22 8/9/82
PAGE 0073

| MT358F10 LABEL | MLP VALUE | FAMILY DEFN | ASSEMBLER REFERENCES | 1.0 | 12:43:22 | 8/ 9/82 | PAGE 0074 |
|-------------------|--------------|----------------|-------------------------|-----------|-----------|-----------|-----------|
| RSTERR | 00BF | 0267 | 1105 | 1316 | | | |
| RSTYPE | 0017 | 0333 | 0762 | | | | |
| RSVERI | 0018 | 0334 | 0544 | 1207 | | | |
| RSWRIT | 000E | 0335 | 0495 | | | | |
| RSYNC | FCA4 | 1016 | 1308 | | | | |
| RSYNCO | FCA7 | 1018 | 0645 | | | | |
| RSYNC2 | FCC8 | 1032 | 1043 | 1045 1049 | | | |
| RSYNC3 | FCDO | 1036 | 1053 | | | | |
| RSYNC4 | FCEB | 1050 | 1047 | | | | |
| RSYNC5 | FCFC | 1060 | 1058 | | | | |
| RSYNC6 | FD02 | 1062 | 1059 | 1064 | | | |
| RSYNC7 | FD27 | 1080 | 1085 | | | | |
| RSYNC8 | FD2C | 1083 | 1079 | | | | |
| RSYNC9 | FD33 | 1088 | 1082 | | | | |
| RSYNCA | FCE4 | 1046 | 1044 | | | | |
| RSYNXT | FCD6 | 1039 | 1036 | | | | |
| RTDL2 | F930 | 0515 | 0510 | 0512 | | | |
| RTDL3 | F932 | 0517 | 0504 | 0558 | | | |
| RTSTAT | F864 | 0412 | 0522 | 0827 | | | |
| SAB | 002D | 0061 | 0062 | 0063 0064 | 0065 | 0066 | 0067 |
| | | | | 0071 0076 | 0378 | | 0068 |
| | | | | | | 0067 | 0070 |
| SCREWD | 0007 | 0355 | | | | | |
| SEQUEN | 0020 | 0258 | 0656 | | | | |
| SETBIT | 0008 | 0222 | 1079 | 1082 1085 | 1090 | 1093 | |
| SLEEP | 0000 | B0008 | 0373 | | | | |
| SN | 00FB | 0212 | 0376 | 0833 | | | |
| SNOTME | FFCE | 1512 | 1485 | | | | |
| STACK | 0003 | 0057 | 0059 | 0368 0910 | | | |
| STACKL | 0013 | 0058 | 0059 | | | | |
| START | 00A0 | B0009 | 1074 | 1251 1466 | | | |
| STATUS | 002D | 0072 | 0406 | 0413 0428 | 0515 0583 | 0596 0738 | 0748 0778 |
| | | | | 0826 0993 | 1201 1207 | 1217 | |
| STFIL2 | FEE7 | 1359 | 0470 | 0716 0722 | | | |
| STFIELD | FEE5 | 1357 | 1527 | | | | |
| STOP | 00FB | 0213 | 0424 | 1199 1225 | | | |
| SYNCDT | 00AA | 0283 | 0852 | | | | |
| TBUS | 0012 | 0356 | 0576 | 1137 1140 | 1144 1147 | | |
| TCATAL | 0008 | 0303 | 0654 | | | | |
| TEMP1 | 002C | 0075 | 0816 | 0821 0822 | 0824 | | |
| TEMP2 | 002B | 0074 | 0548 | 0551 | | | |
| TEMP4 | 0029 | 0073 | 0698 | 0710 | | | |
| TEOF2 | FEC6 | 1335 | 1328 | 1332 | | | |
| TEST | 0004 | A0007 | 0382 | 0883 1023 | | | |
| TFLOAT | 0016 | 0357 | 0431 | 0502 0536 | 0571 0924 | 1400 1402 | 1416 1420 |
| | | | | 1453 | | | |
| TFORMA | 0001 | 0304 | 0634 | 0664 | | | |
| TIME | 0002 | A0003 | 0435 | 1029 1067 | 1248 | | |
| TIMER | 0003 | A0006 | 0373 | 0436 1018 | 1074 1226 | 1251 1459 | 1466 |
| TIMEX | 000A | 0358 | 0832 | 1016 1198 | 1235 | | |
| TNCATA | 0008 | 0305 | 0635 | | | | |
| TNOPFN | 0004 | 0306 | 0655 | | | | |
| TOPEN | 0004 | 0307 | 0667 | | | | |
| TSTB10 | FEC8 | 1340 | 1340 | 1526 | | | |
| TSTB4B | FFBE | 1501 | 1501 | 1525 | | | |
| TSTR8A | FED1 | 1344 | 1344 | 1524 | | | |
| TSTB8B | FEDA | 1348 | 1348 | 1523 | | | |
| TSTBIT | FFBD | 1470 | 1470 | 1529 | | | |

| LABEL | VALUE | DEFN | REFERENCES | | | |
|--------|-------|-------|------------|------|------|------|
| TSTEOF | FEB2 | 1326 | 1521 | | | |
| TSTEOT | FC73 | 0985 | 1528 | | | |
| TSTI | 0080 | 0272 | 0449 | 0660 | 0763 | |
| TSTIU | 0040 | 0273 | 0562 | 0755 | | |
| TSTNA | 0000 | 0274 | 0657 | | | |
| TSTOU | 0080 | 0275 | 0493 | 0689 | | |
| UNSUPP | F86C | 0419 | 0397 | 1483 | 1484 | 1486 |
| WAFER | 000A | A0009 | 0392 | 0985 | 1036 | 1222 |
| WAKEUP | FF7F | 1457 | 0432 | 0433 | 0433 | 1539 |
| WBYTE | FBC1 | 0860 | 0866 | 1006 | | |
| WCHKSM | FC4E | 0959 | 1007 | | | |
| WDIREC | FC86 | 0998 | 0454 | 0792 | 0907 | |
| WDL | FC25 | 0929 | 0503 | | | |
| WE | 00F9 | 0214 | | | | |
| WEOT2 | FC78 | 0987 | 0985 | | | |
| WEOT3 | FC7A | 0988 | 0981 | | | |
| WEOT4 | FC82 | 0993 | 0990 | | | |
| WGHOST | FBB3 | 0850 | 0847 | | | |
| WHELP | FBD3 | 0876 | 0920 | 0921 | | |
| WLASTO | FC62 | 0974 | 0968 | | | |
| WMEM | FC99 | 1006 | 0708 | | | |
| WNIB | FC3B | 0944 | | | | |
| WNIBO | FC41 | 0947 | 0954 | | | |
| WNIB3 | FC47 | 0952 | 0946 | | | |
| WNXHSK | FF3C | 1416 | 1425 | 1534 | | |
| WNXPA2 | FF43 | 1420 | 0521 | 0586 | | |
| WP | 0004 | 0219 | 0392 | | | |
| WPERET | 009B | 0269 | 0391 | | | |
| WRBIT1 | 0006 | C0016 | 1245 | | | |
| WRBIT2 | 0002 | C0017 | 1246 | | | |
| WSYNC | FB8D | 0832 | 0501 | 0706 | | |
| WSYNC0 | FB91 | 0835 | 1004 | | | |
| WSYNC2 | FBA4 | 0845 | 0842 | | | |
| WSYNC3 | FBD6 | 0880 | 0850 | 0853 | 0886 | |
| WSYNC4 | FBE0 | 0884 | 0882 | | | |
| WSYNC5 | FBE8 | 0890 | 0836 | 1237 | | |
| WSYNC6 | FBF1 | 0894 | 0897 | | | |
| WTERR | F86E | 0423 | 1536 | | | |
| XCATA2 | FB56 | 0804 | 0799 | 0801 | | |
| XCATAL | FB48 | 0798 | 1509 | | | |
| XCL0D0 | F9C3 | 0607 | 0604 | | | |
| XCL0D1 | F9DC | 0620 | 0621 | | | |
| XCL0D3 | F9F8 | 0633 | 0630 | | | |
| XCL0D4 | FA32 | 0658 | 0656 | | | |
| XCL0D5 | FA38 | 0661 | 0634 | 0655 | | |
| XCL0D6 | FA3E | 0664 | 0637 | 0654 | 0660 | 0661 |
| XCL0D7 | FA4D | 0669 | 0664 | 0667 | | |
| XCL0D8 | FA4F | 0671 | 0682 | | | |
| XCL0D9 | FA34 | 0660 | 0657 | | | |
| XCL0D0 | F9EF | 0628 | 1489 | | | |
| XCL0D0 | F9CA | 0611 | 1481 | | | |
| XCL0D0 | F9F2 | 0630 | 0625 | 1488 | | |
| XCL0D0 | F9BC | 0603 | 1475 | | | |
| XCL0DX | F9C1 | 0605 | 0611 | 0613 | 0617 | |
| XCL0RS | F8AF | 0455 | 0449 | | | |
| XCLOSE | F89E | 0448 | 1493 | | | |
| XCL0WD | F8AC | 0454 | 0468 | 0478 | | |

MT358F10 MLP FAMILY ASSEMBLER 1.0
LABEL VALUE DEEN REFERENCES

12:43:22 8/9/82

8/9/82

PAGE 0076